VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2020 Artur Shestopalov

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

ROZPOZNÁVÁNÍ BEZPEČNOSTNÍCH INCIDENTŮ V SÍTÍCH TYPU LPWA
RECOGNITION OF SECURITY ISSUES IN LPWA TYPE NETWORKS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Artur Shestopalov**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Ondřej Pospíšil**
SUPERVISOR

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Artur Shestopalov ***ID:*** 195170
Ročník: 3 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Rozpoznávání bezpečnostních incidentů v sítích typu LPWA

POKYNY PRO VYPRACOVÁNÍ:

Student se v práci zaměří na sítě LPWA (Low Power Wide Area) jako například Sigfox, NB-IoT či LoRaWAN. Provede analýzu těchto technologie a vybere jednu technologii, na kterou se dále blíže zaměří. V rámci vybrané technologie bude provedena vektorizace útoků a tedy i shrnutí (popsání) bezpečnostních hrozeb/zranitelností jednotlivých částí sítě. V rámci praktické části bude zprovozněna funkční komunikace od koncového prvku až po server (privátní síť). Následně bude navržena regulérní aplikace (komunikace) společně s neregulérní komunikací (bezpečnostními incidenty) odpovídající provedené analýze zranitelností (vektorizaci útoků). Bude vytvořen systém detekce pro neregulérní provoz, který na základě parametrů sítě, ale i dalších znaků, bude schopen rozpoznávat vybrané bezpečnostní incidenty v jednotlivých částech sítě. Systém detekce bude obsahovat grafické rozhraní tak, aby bylo možné statistiky popř. výsledky analýzy reprezentovat.

DOPORUČENÁ LITERATURA:

[1] MADHUMITHA, M.; SINGH, Bhupendra Pratap. A survey on LPWAN technologies in content to IoT applications. 2019.

[2] BEMBE, Mncedisi, et al. A survey on low-power wide area networks for IoT applications. Telecommunication Systems, 2019, 71.2: 249-274.

Termín zadání: 3.2.2020 ***Termín odevzdání:*** 8.6.2020

Vedoucí práce: Ing. Ondřej Pospíšil

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Bakalářská práce se zabývá problematikou rozpoznávání bezpečnostních incidentů v sítích typu LPWA (Low Power Wide Area). Práce obsahuje informace o nejpoužívanějších LPWA technologiích, zkoumá se jejich bezpečnostní architektura a možné hrozby. Součástí práce je podrobný popis protokolů LoRaWAN (Long Range Wide Area Network) včetně architektury, možných útoků, rozlišení bezpečnostní architektury v nové a staré verzi. Práce obsahuje návrh a realizaci dvou scénářů bezpečnostních incidentů a jejich následnou detekci, jedná se o jamming a výpadek brány. V rámci detekce byla vytvořena webová aplikace, která je schopna v reálném čase detekovat bezpečnostní incidenty na základě střední hodnoty RSSI (Received Signal Strength Indicator) a očekávaného času přenosu zprávy.

KLÍČOVÁ SLOVA

Chirpstack, detekce anomálií, IoT, jamming, LoRa, LoRaWAN, Loriot, LPWAN

ABSTRACT

The bachelor's thesis deals with the issue of recognizing security incidents in LPWA (Low Power Wide Area) networks. Thesis contains information about the most used LPWA technologies, examines their security architecture and possible threats. Part of the work is a detailed description of LoRaWAN protocols, including architecture, possible attacks, distinguishing security architecture in the new and old versions. Work contains implementation of two security incident scenarios and their subsequent detection: jamming and gateway disconnection. As part of the detection, was created a web application that is able to detect real-time security incidents based on the average value of RSSI (Received Signal Strength Indicator) and the expected time of message transmission.

KEYWORDS

Anomaly detection, Chirpstack, IoT, jamming, LoRa, LoRaWAN, Loriot, LPWAN

SHESTOPALOV, Artur. *Rozpoznávání bezpečnostních incidentů v sítích typu LPWA*. Brno, 2020, 68 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav elektroenergetiky. Vedoucí práce: Ing. Ondřej Pospíšil

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Rozpoznávání bezpečnostních incidentů v sítích typu LPWA“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing.Ondřeji Pospíšilovi za odborné vedení, konzultace, trpělivost, pozornost a podnětné návrhy k práci.

Obsah

Seznam obrázků

Seznam tabulek

Seznam výpisů

Úvod

S explozivním růstem technologií IoT (Internet věcí) lze v současné době nalézt rostoucí počet praktických aplikací v mnoha oblastech, včetně bezpečnosti, sledování majetku, zemědělství, inteligentního měření, inteligentních měst a inteligentních domů [1]. IoT nabízí pokročilejší služby, technologie propojuje různé subsystémy přes Internet. Typická zařízení internetu věcí zahrnují senzory a mikročipy, které mohou shromažďovat nebo přenášet informace prostřednictvím Internetu. Obrovská integrace IoT zařízení vyvolává náročné otázky: škálovatelnost sítě a hlavně její bezpečnost.

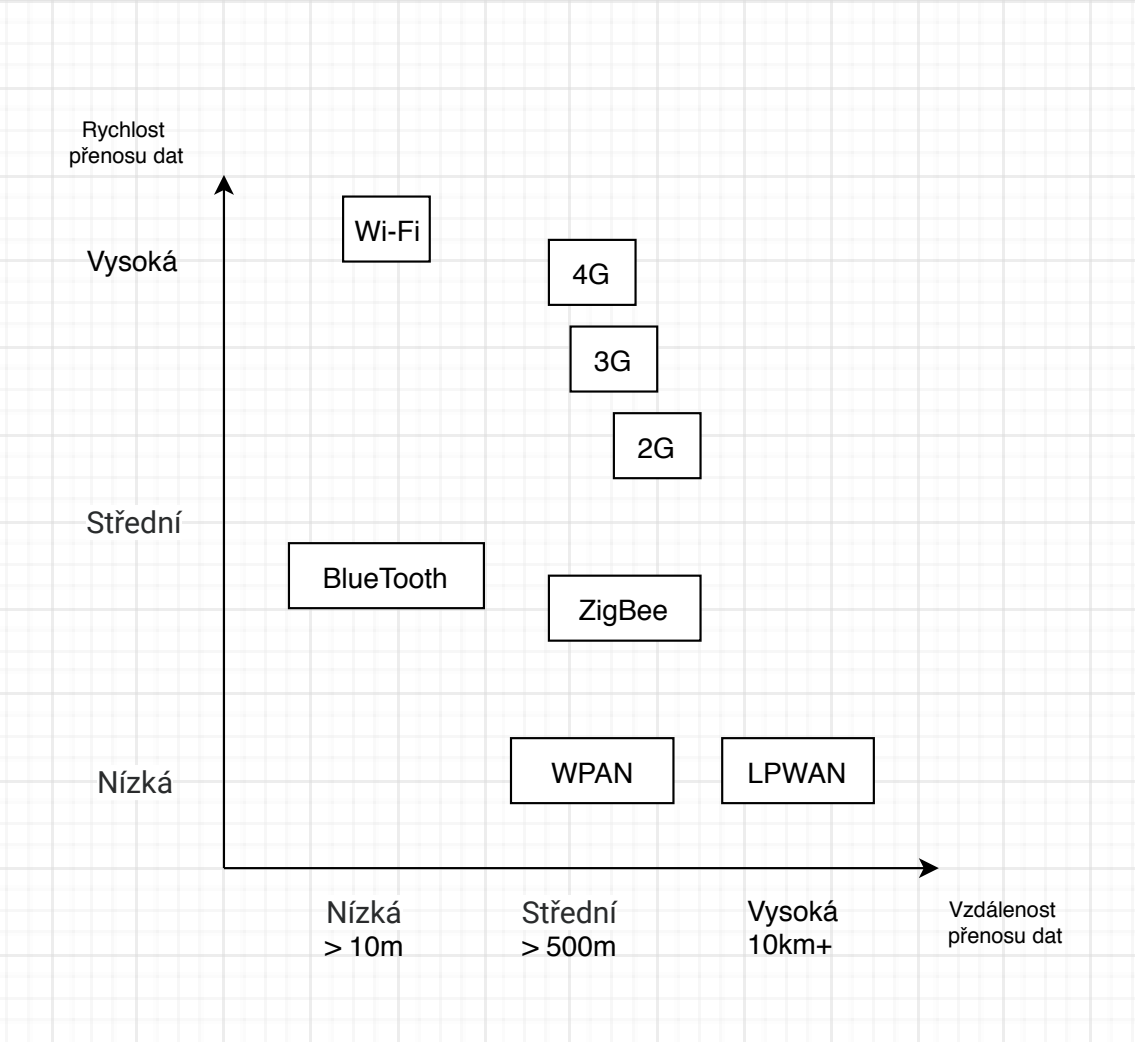
Tato práce cílí zejména na zkoumání IoT problematiky, analýzu různých druhů LPWA (Low Power Wide Area) protokolů, jako jsou například NB-IoT, LoRaWAN (Long Range Wide Area Network) a Sigfox, se zaměřením na jejich bezpečnostní architekturu. Velký důraz je kladen na simulaci bezpečnostních incidentů a jejich následnou detekci a odhalování. V praktické části bude vytvořena vlastní experimentální síť a zprovozněn server, poté budou navrženy scénáře na testování bezpečnostních incidentů a možnosti jejich detekce. Následně budou provedeny dva bezpečnostní incidenty (jammming a výpadek brány) a také bude vytvořena aplikace, která bude schopná tyto incidenty rozpoznávat.

Práce se skládá ze čtyř částí. První část se zaměřuje na analýzu, využitelnost a porovnání nejpoužívanějších LPWA (Low Power Wide Area) protokolů. Druhá část bude obsahovat podrobnější rozbor protokolů LoRaWAN, jeho infrastrukturu, bezpečnost, jednotlivé hrozby a bezpečnostní rozdíl ve verzích 1.0 a 1.1. Ve třetí části bude provedena analýza anomálií a bezpečnostních incidentů v síti a následně návrh scénářů detekce. Poslední část bude praktická a zaměří se na zprovoznění sítě a serverů, simulaci bezpečnostních incidentů a následné vytvoření aplikace na detekci incidentů.

1 LPWA sítě

Nízkoenergetické širokopásmové sítě LPWAN (Low Power Wide Area Network) postupně získávají stále větší popularitu zejména v aplikacích IoT, kde je lze využít v průmyslu. Nabízejí energeticky řešení a jsou schopné komunikovat na velké vzdálenosti, viz obr. 1.1. Mohou komunikovat v těžko dosažitelných místnostech, a to díky malým rozměrům koncových zařízení a odolnému signálu. Používají se v různých oblastech: zdravotnictví, inteligentní domy, strojírenský a zpracovatelský průmysl, obchodování a mnoho dalších.

Používají ke své komunikaci topologii typu hvězda [2]. Tento přístup zjednodušuje implementaci sítě a poskytuje obrovskou výhodu pro inteligentní síť – porucha koncového bodu nemá žádný vliv na výkon ostatních zařízení.



Obr. 1.1: Porovnání LPWAN s jinými sítěmi.

1.1 Požadavky na LPWA sítě

- Velký dosah – 5 až 10 km ve městě a 10 až 30 km ve venkovských oblastech. Závisí většinou na počtu překážek a úrovni šumu, který je ve venkovských oblastech mnohem menší než ve městě.
- Nízké ceny koncového zařízení a komunikačního modulu [3].
- Použití licenčního nebo bezlicenčního pásma. Bezlicenční pásmo v Evropě funguje na frekvenci 868 mHz a v USA 915 mHz. Rozdíl je v potřebách uživatele. Bezlicenční pásmo – neplacené, používá se pro vytvoření privátních sítí, licenční naopak, když je potřeba použít nějaké hotové řešení a v tomto případě zaplatit licenční poplatek.
- Nízká spotřeba energie – zařízení pro svou činnost nepotřebuje být vždy zapnuté a čerpat obrovské množství elektřiny, po odeslání dat se může i na nějaký čas vypnout.

1.2 Využitelnost LPWA sítí

Technologie LPWA slouží většinou k monitorování infrastruktury (včetně různých případů měření, např. monitorování spotřeby plynu, elektřiny, vody). Tyto sítě nejsou vhodná pro aplikace, kde je vyžadována vysoká rychlost přenosu dat a vysoká spolehlivost.

V dnešní době se LPWA sítě používají pro různé účely:

- Zdravotnictví – sledování pacientů v nemocnicích a domovech pomocí jediného systému, ten může sledovat různé hodnoty na zařízeních a v případě potřeby hlásit události.
- Průmysl – sledování stavu vzduchu ve výrobnách nebo továrnách z důvodu výskytu škodlivých látek.
- Chytrá města – měření a monitorování energetických sítí, parkování, inteligentní osvětlení, kontrola dopravy a mnoho dalších.
- Chytré domy – různá signalizační zařízení, inteligentní měřiče, systém osvětlení, senzory na únik plynu atd.

Technologie LPWAN má řadu výhod, které umožňují implementovat řešení rychleji a levněji než obvyklými způsoby. Přínos je zjevný zejména u velkých projektů, kde je vyžadováno velké množství samostatných zařízení a široké pokrytí území.

1.3 Protokoly

V dnešní době se používá hodně protokolů včetně: RPMA (Random Phase Multiple Access), LTE-M, EC-GMS-IoT, Weightless, LoRaWAN, SigFox, NB-IoT a další.

Tato práce se zabývá podrobněji LoRaWAN, SigFox a NB-IoT, protože tyto protokoly jsou v současnosti velmi rozšířené. Sigfox je nejstarší protokol, existuje od roku 2010, má obrovské pokrytí v Evropské unii a skoro po celém světě, je to komerční řešení. NB-IoT je nejpopulárnější technologie v současné době, poskytuje řešení založené na 4G infrastruktuře, díky tomu má obrovské pokrytí a zvýšenou úroveň bezpečnosti. LoRaWAN je také velice populární technologie a na rozdíl od SigFox a NB-IoT dovoluje vytvoření privátní sítě, což může snížit náklady na síť a zvyšuje flexibilitu během vytvoření sítí.

1.3.1 LoRaWAN

Sítě využívající protokol LoRaWAN používají pro svou činnost nelicencovaná frekvenční pásma. Jedna základnová stanice může obsluhovat několik desítek tisíc zařízení, což je způsobeno velkým pokrytím signálem a vysokou odolností vůči šumu, přičemž nabízí minimální požadavky na infrastrukturu a optimální životnost baterie. O podporu a další vývoj LoRaWAN se starají firmy jako IBM, Semtech, Cisco, a podobně.

LoRa je technika modulace a funguje na fyzické vrstvě, LoRaWAN je protokol pro nízkoenergetické sítě vyvinutý společností LoRa Alliance pro LPWAN. Modulace LoRa je založena na technologii rozprostřeného spektra a techniky, která umožňuje rozšířit spektra – CSS (Chirp Spread Spektrum) s integrovanou korekcí chyb FEC (Forward Error Correction).

Uzly sítě LoRaWAN mohou provádět různé funkce, například: měření, řízení a monitorování. Pomocí protokolu LoRaWAN tyto uzly komunikují s bránou. Data z uzlů jsou přenášena v obou směrech – na server a zpátky. Uzly pracují v režimu přenosu pouze po krátkou dobu, poté se otevře časové okno pro příjem dat. Po nějakém čase uzly přecházejí do stavu spánku pro šetření energie.

1.3.2 SigFox

Sigfox patří k francouzské společnosti a zároveň je provozovatelem sítě, je to velmi úspěšná technologie a používá se aktuálně v 45 zemích [4]. Sigfox pokrývá 94 % území České republiky, na to bylo použito 320 vysílačů, za měsíc přes síť Sigfox v Česku prochází kolem 25 milionů zpráv [5]. Jednou z největších výhod sítě Sigfox je to, že nemusíte vytvářet síť, spravovat brány atd. Po registraci v síti a placení speciálního modulu se k ní můžete připojit, odesílat a přijímat data. Čipy Sigfox lze použít pouze v místě, kde je nastavena stávající síť. Díky tomu, že Sigfox používá bezlicenční ISM (industrial, scientific and medical) pásma (není placená) a ke stavbě sítí využívá mobilního operátora T-Mobile (v Česku). Sigfox modul stojí 2 dolary [6], což je ve srovnání s konkurenty, považováno za nízkou cenu.

Většinou se u technologií LPWAN na fyzické vrstvě používají dvě hlavní komunikační techniky, UNB (Ultra-Narrow Bandwidth) a SS (Spread Spectrum). Sigfox používá UNB modulace.

Technika UNB se používá k přenosu signálu s velmi malou šířkou pásma (méně než 1 kHz) a většinou se hodí pro malý uplink dat [7]. Koncentrace energie na tak úzkou frekvenční šířku pásma vede ke snížení požadavků na energii a díky tomu pak přeskočí úroveň šumu, což umožňuje komunikaci na velké vzdálenosti s menším přenosovým výkonem [8].

Je to poměrně stará technika, ale byla obnovena pro komunikaci IoT a M2M díky pokroku ve sféře IoT.

V závislosti na prostředí přijímají stanice zprávy na vzdálenosti 10 až 50 km. Pracovní doba v Evropě je omezena a dovoluje pouze 36 přenosových sekund za hodinu. Zařízení Sigfox může odesílat šest zpráv za hodinu s maximálním zatížením – 12 bajtů na zprávu, a to kvůli tomu, že modulace UNB má omezenou přenosovou rychlost – 100 bps [9]. Navzdory těmto omezením Sigfox zůstává užitečným standardem LPWAN pro různé aplikace v IoT, které nejsou časově kritické, např. snímání kvality vzduchu, sledování vodoměrů apod.

1.3.3 Nb-IoT

NB-IoT má oproti jiným LPWA sítím řadu výhod. Používá výhody 4G: infrastrukturu, dosah a bezpečnost. Tato technologie má vynikající pokrytí a široký rozsah. NB-IoT může vzájemně propojit miliony zařízení (připojí až 100 tisíc zařízení k jedné základnové stanici). Tato technologie využívá existující síť a licencované frekvence, jako jsou 3G a 4G, což znemožňuje jiným uživatelům zasahovat do sítě a zvyšuje úroveň bezpečnosti. Neexistuje také žádné zákonné omezení pro použití dat.

Je použita modulace rozprostřeného spektra. Data jsou rozdělena a současně vysílána na co nejvíce frekvencích v určitém pásmu. Tento způsob poskytuje vyšší přenosové rychlosti než FHSS (Frequency-Hopping Spread Spectrum), který používá omezený počet frekvencí.

NB-IoT využívá existující funkce LTE, lze používat stejný hardware a také sdílet spektrum bez problémů s koexistencí. To umožňuje levné a rychlé použití NB-IoT pomocí stávající infrastruktury [10]. NB-IoT je schopen pracovat při poruchách a umožňuje šetřit baterie. Používá se většinou na přenos krátkých zpráv a neslouží k přenosu audio-video obsahu nebo velkých souborů. Očekává se, že NB-IoT bude součástí standardu 5G (Fifth Generation) [11].

Provoz NB-IoT vyžaduje minimální šířku pásma – 180 kHz, maximální rychlosti na stahování a nahrávání dat dosahují 250 kbitů [12]. Umožňuje využití QoS (Quality of Service), což dále dovoluje stanovit pevnou šířku pásma a zabránit situacím,

kdy jedno zařízení bude využívat všechny prostředky sítě.

1.4 Bezpečnostní architektura v Sigfox a NB-IoT sítích

Pro zabezpečený přenos dat z cloudu na zařízení **Sigfox** používá metodu ověřování odesílatele na základě bezpečnostního klíče spojeného s ID zařízení – HMAC (Hash-Based Message Authentication Code) [13]. Tato metoda je specifickým příkladem MACu (Message Authentication Code), s tím rozdílem, že koncová zařízení používají hašovací funkce a tajný klíč na vytvoření podpisu [14]. Podpis je jedinečný pro každou zprávu a používá se pro ověření a autentizaci odesílatele. Aby se zabránilo opakovanému vysílání rámců, podpis obsahuje pořadové číslo (sequence number). Každá zpráva je posílána třikrát a na třech různých frekvencích, což pomáhá chránit rádiové snímky před krádeží [15].

NB-IoT stejně jako Sigfox používá MAC, jenom s tím rozdílem, že NB-IoT využívá klíče symetrických šifer na vytvoření zabezpečené zprávy. Používá: NULL, SNOW3G [16], AES [17] (Advanced Encryption Standard), ZUC [18].

1.5 Porovnání jednotlivých protokolů

Tab. 1.1: Technické rozdíly SIGFOX, LoRaWAN a NB-IOT.

	Sigfox	**LoRaWAN**	**NB-IoT**
Modulace	BPSK	CSS	QPSK
Frekvence	Pásma bez licencí ISM 868 MHz v Evropě	Pásma bez licencí ISM 868 MHz v Evropě	Licencovaná pásma LTE
Šířka pásma	100 Hz	250 kHz a 125 kHz	200 kHz
Max. rychlost	100 bps	50 kbps	200 kbps
Max. zpráv	140 (UL), 4 (DL)	Neomezen	Neomezen
Rozsah	10 km (město) 40 km (vesnice)	5 km (město) 20 km (vesnice)	1 km (město) 10 km (vesnice)
Šifrování	ve výchozím stavu není podporováno	AES 128 b	LTE zabezpečení
Obousměrnost	Limitovaná/ half-duplex	Ano/ half-duplex	Ano/ half-duplex
Standard	Sigfox a ETSI	LoRa-Alliance	3GPP

LoRaWAN

Protokol LoRaWAN je na rozdíl od NB-IoT a Sigfox vhodný na vytvoření privátní sítě, a pro to je potřeba koncové zařízení k odeslání dat, síťovou bránu a nakonfigurovaný síťový a aplikační server (často jsou součástí jedné entity).

K výhodám patří: delší životnost baterie než u NB-IoT [19], schopnost pracovat v pohybu, což je užitečné hlavně pro logistické účely, možnost vytvoření privátní sítě.

Nevýhody: má nižší rychlost přenosu dat a má delší dobu zpoždění než NB-IoT. Vyžaduje bránu (což je v některých případech výhodou), LoRaWAN na rozdíl od NB-IoT používá nelicencované spektrum (omezuje objem a frekvenci provozu).

Sigfox

Sigfox je komplexní síť. Má nejnižší náklady na rádiové moduly (< 2 €, ve srovnání s 3–5 € pro LoRa a > 20 € pro NB-IoT) [20]. Sigfox je pouze uplink. Ačkoli je možný downlink, má jiný rozpočet na propojení a je velmi omezený.

Výhody: funguje dobře pro jednoduchá zařízení, která přenášejí zprávy jen zřídka, protože odesílá velmi malé množství dat velmi pomalu. Spotřebuje malé množství energie. Podporuje širokou oblast pokrytí.

Nevýhody: není nasazena všude (většinou v Evropě), takže v současné době nebude pro některé země fungovat. Mobilita je poměrně obtížná, na rozdíl od LoRaWAN.

NB-IoT

Kvůli tomu, že NB-IoT je mobilní bezdrátová technologie, jsou čipy složitější. To znamená, že uživatelé získají vysokou úroveň výkonu s celulárním propojením, ale spotřeba energie bude mnohem vyšší. NB-IoT je určen ke zpracování malého množství dat (několik desítek nebo stovek bajtů denně).

Hlavní výhodou oproti Sigfox a LoRaWAN je mnohem rychlejší modulační rychlost, která dokáže přenést více dat (má rychlejší dobu odezvy než LoRaWAN a může zaručit lepší kvalitu služeb), kromě toho zajišťuje dobrou kvalitu pokrytí 4G, a to dokonce v hustých městských částech. Byl vytvořen pro méně časté komunikační účely než Sigfox a LoRaWAN.

Nevýhody: není podporován roaming, na rozdíl od Sigfox [21].

2 LoRaWAN

Tento protokol bude v práci popsán detailněji než ostatní, a to kvůli tomu, že je ideální pro zkoumání a návrh vlastní privátní sítě. Na obr. 2.1. jsou výsledky vyhledávání jednotlivých technologií v „Google Scholar“ za poslední roky. Z toho je vidět, že nejpopulárnější z nich je NB-IoT, ale na rozdíl od LoRaWAN ji nelze provozovat jako vlastní privátní síť (používá se jenom pomocí stávající infrastruktury a využívá se licenční pásmo).

Pro návrh sítě LoRaWAN je potřeba pouze koncové zařízení, brána, nakonfigurovaný síťový a aplikační server.

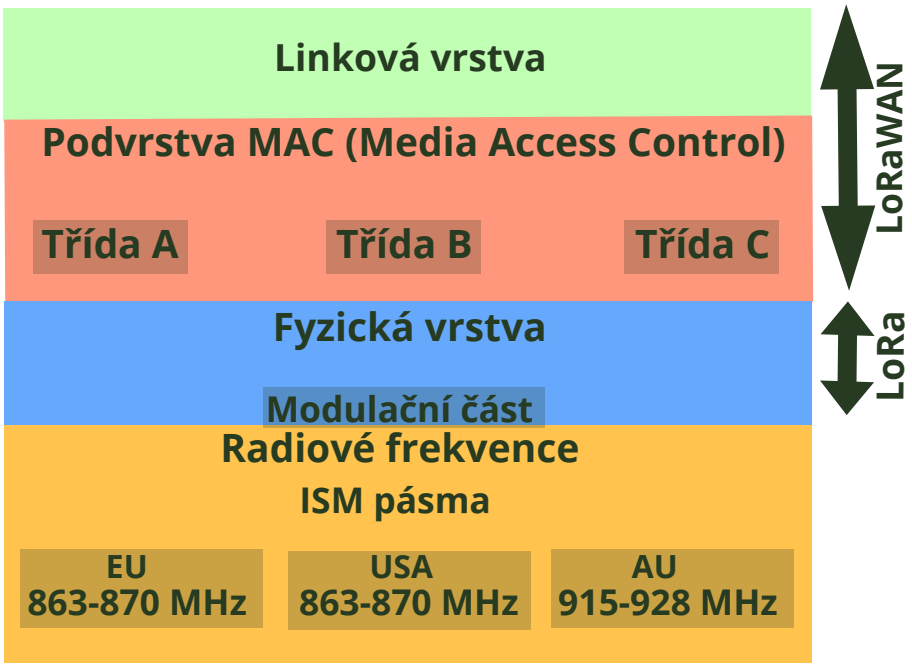
Obr. 2.1: Porovnání počtu článků.

2.1 Architektura protokolu LoRaWAN

2.1.1 Fyzická vrstva

Strukturu protokolu LoRa (Long Range) je vidět na obr. 2.2. Hlavní rozdíl mezi LoRa a LoRaWAN spočívá v tom, že LoRa je technika bezdrátové modulace, LoRaWAN je komunikační protokol a systémová architektura. LoRa nabízí výrazně delší dosah než tradiční telekomunikace, nachází se na fyzické vrstvě modelu ISO/OSI.

Používá modulaci s rozprostřeným spektrem (CSS). Ve 40. letech 20. století byla technika běžně používána armádou pro radarové aplikace a v bezpečných komunikačních aplikacích [22].



Obr. 2.2: Vrstvy LoRaWAN protokolu.

Používá různé faktory šíření SF (Spreading Factor) od SF7 do SF12. Faktor šíření je délka chirpu. SF7 je nejkratší doba přenosu, SF12 nejdelší. Každý krok zdvojnásobuje čas přenosu stejného množství dat. Tento komunikační mechanizmus poskytuje zvýšenou odolnost proti rušení. Uzly umístěné v blízkosti brány budou mít vyšší datovou rychlost ve srovnání s uzly na větší vzdálenosti, které jsou nuceny zvýšit faktor šíření, což zapříčiní snížení datové rychlosti.

LoRa umožňuje snížit náklady na zařízení, prodloužit životnost baterií na zařízeních, zlepšit kapacitu sítě a podporovat velké množství zařízení. Ideálně se hodí na komunikaci mezi uzly pomocí P2P (peer-to-peer).

2.1.2 Linková vrstva (MAC vrstva)

Nad LoRa se nachází podvrstva MAC (Media Access Control). Poskytuje mechanizmy adresování a řízení přístupu pro kanály, které umožňují více terminálům nebo přístupovým bodům vzájemně komunikovat, a emuluje duplexní logický komunikační kanál ve vícebodové síti. Hlavním cílem je zabránit kolizím a usnadnit přenos paketů mezi dvěma body. Ke kolizím může docházet, když dva a více terminálů přenáší data současně. Může to vést k rušení komunikace.

Kolizím lze zabránit pomocí metod přístupu k médii:

- Deterministické – přístup uzlu k přenosovému médiu v předem stanoveném pořadí (každý uzel musí dostat přístup do sítě v určitém časovém intervalu). Jed-

nou z variant deterministického přístupu je řízený přístup, ten spočívá v tom, že nějaká řídící stanice bude kontrolovat čas vysílání každého uzlu.

- Stochastické – uzly mají čekat na volná místa (nepoužívají časové intervaly), k jeho variantám patří:
  1. CSMA/CD (Carrier Sense Multiple Access/Collision Detection) – stanice bude naslouchat a čekat na moment, kdy žádná jiná stanice nevysílá a poté začne vysílání svých datových rámců
  2. ALOHA – stanice posílá rámec v libovolný okamžik času. V případě, kdy žádné potvrzení nepřijde, bude to znamenat, že došlo ke kolizi a následně přepošle stejný rámec. LoRaWAN používá ALOHA.

2.1.3 Síťová infrastruktura LoRaWAN

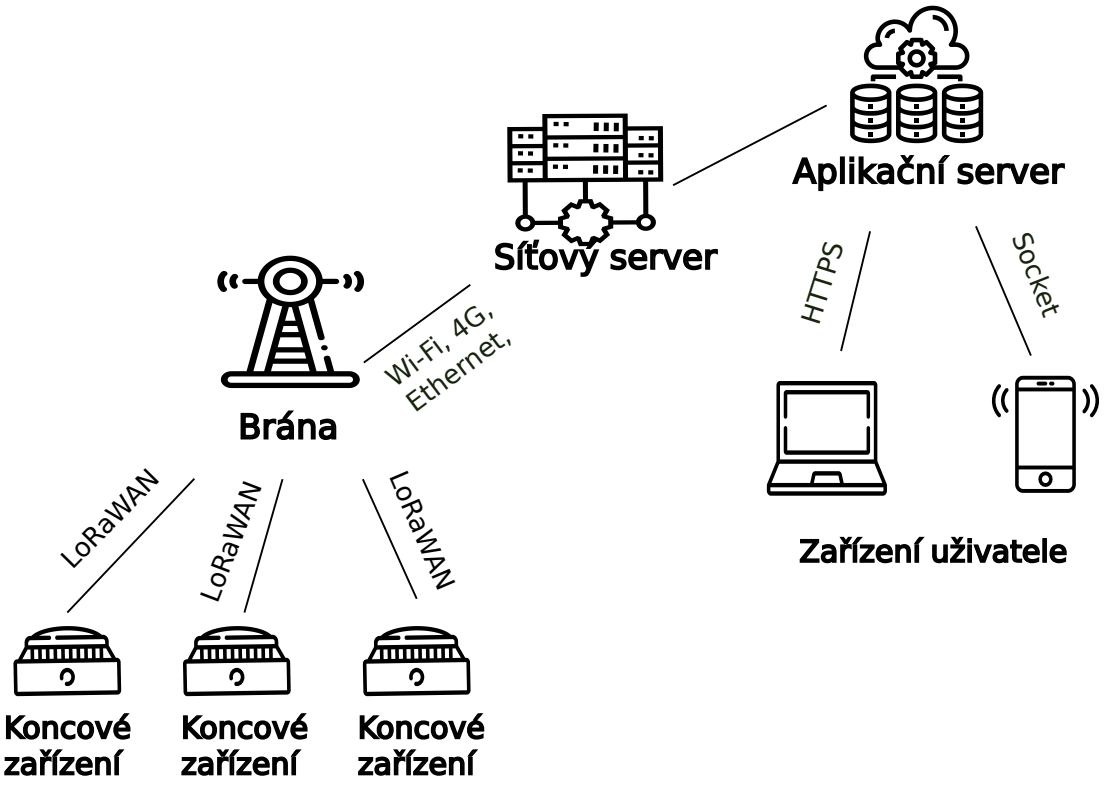
LoRaWAN funguje na linkové vrstvě a je postaven na fyzické vrstvě s modulací LoRa. Zahrnuje v sobě také síťovou vrstvu. Úkolem LoRa je zajišťování komunikace na velké vzdálenosti, zatímco LoRaWAN definuje architekturu systému pro síť a komunikační protokol. Spojení brány a síťového serveru je zajištěno převáděním signálu do transparentní podoby a přenáší se pomocí různých technologií, např. WiFi, Ethernet, 4G, EDGE. Jedna brána může mít spoustu spojení. Síťové servery spravují data a provádějí bezpečnostní kontroly nebo filtrují redundantní přijaté pakety. Poté jsou předávány do aplikačních serverů a dále pomocí HTTPS spojení nebo WebSoketu, zařízení uživatele nebo databáze obdržují data z API serveru, viz obr. 2.3.

Zařízení v síti LoRaWAN komunikují pouze pokud mají data připravená k odeslání (jsou asynchronní), ostatní čas se nacházejí ve stavu spánku, což umožňuje šetřit baterie, která je navíc optimalizována pomocí adaptivní rychlosti přenosu dat [23]. To znamená, že přenosová rychlost se spravuje zvlášť pro každé koncové zařízení.

Kromě adaptivní rychlosti přenosu dat se brána používá coby vícekanálový vysílač a přijímač, ten umožní dosáhnout vysoké kapacity sítě a následně přijímat zprávy na různých kanálech. Síť LoRaWAN ke své činnosti nepotřebuje velkou infrastrukturu a jestliže je potřeba zvýšit kapacitu, je možné nainstalovat více bran a tím zmenšit zatížení na jiné brány [23].

LoRaWAN samozřejmě umožňuje jedné bráně pokrývat rozsáhlé oblasti. Pokrytí v městských oblastí může dosahovat 6 km [24]. Ve vesnicích, kde je počet domů a jejich výška menší, pokrytí dosahuje 10 km [25]. Existují různé typy koncových zařízení, které slouží různým aplikacím a mají několik požadavků. Aby bylo možné vyhovět celé řadě koncových aplikací, má síť LoRaWAN různé třídy zařízení. Tyto třídy závisí na síťové komunikaci nebo životnosti baterie: [26].

1. Třída A (obousměrná koncová zařízení): zařízení třídy A používají obousměrnou komunikaci, což znamená, že každý uplink následují dvě přijímací okna na stahování.
2. Třída B (obousměrná koncová zařízení s naplánovanými přijímacími sloty): zařízení třídy B otevírají okna pro přijímání jenom v naplánovaných časech.
3. Třída C (obousměrná koncová zařízení s maximálním časem pro příjem): zařízení třídy C mohou přijímat zprávy kdykoli.



Obr. 2.3: LoRaWAN architektura verze 1.0.

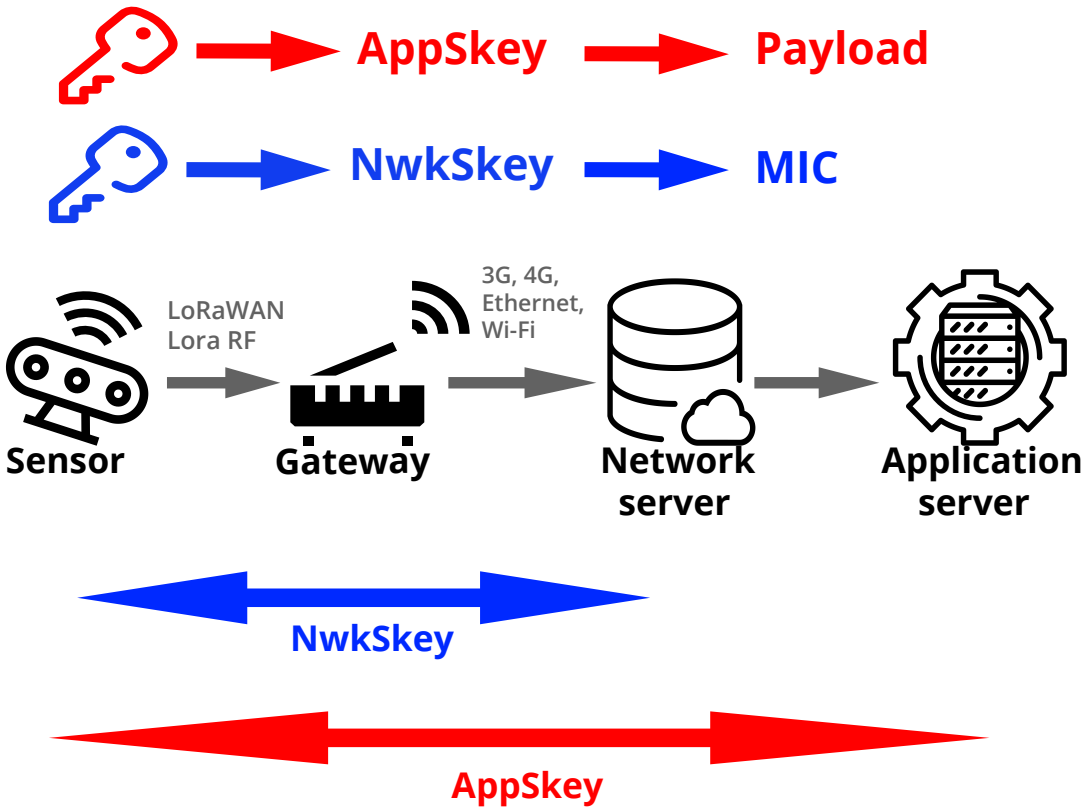
2.2 Bezpečnost

Bezpečnost se liší podle různých druhů aktivací koncového zařízení (aby začal proces vysílání, je potřeba určitým způsobem aktivovat koncový uzel). Existují dvě možnosti: OTAA (Over-The-Air Activation) a ABP (Activation By Personalization).

V případě **OTAA** síťový server a zařízení používají aplikační klíč – AppKey při požadavku na připojení. AppKey je v podstatě symetrická šifra AES-128-bit

v módu CTR (Counter). Z aplikačního klíče jsou odvozeny dva další klíče: aplikační klíč (AppSKey) a síťový klíč relace (NwkSKey), viz obr. 2.4. NwkSkey je sdílen jak koncovým zařízením, tak i síťovému serveru. Síťový server pomocí NwkSkey vygeneruje MIC (Message Integrity Code), kterým pak bude vytvořen zvláštní podpis pro každé zařízení a zajištěna integrita dat. AppSKey se používá pro šifrování nebo dešifrování payloadu a aplikuje se mezi koncovým zařízením a aplikačním serverem. Každá zpráva o žádosti na připojení obsahuje AppEUI, DevEUI a DevNonce zařízení. AppEUI je unikátní ID, který identifikuje vlastníka koncového uzlu, DevEUI je 64bitový identifikátor pouze koncového uzlu. Tyto hodnoty se podepisují 4bajtovým kódem MIC. DevNonce je náhodná hodnota, která je sledována síťovým serverem a je použita k odmítnutí jakékoli žádosti o navázání spojení s neplatnou hodnotou nonce. Celý tento mechanizmus zabraňuje opakovaným útokům.

ABP nepoužívá spojovací zprávy. Před aktivací jsou koncovému zařízení přiřazeny jedinečné parametry – DevAddr, NwkSkey a AppSkey a jsou uloženy na serveru. Když se koncové zařízení pokouší komunikovat se serverem, odešle zprávy přímo do něj. Tyto zprávy jsou šifrovány a podepsány, takže zprávu může číst pouze odpovídající síťový server.



Obr. 2.4: Bezpečnostní architektura verze 1.0.

2.2.1 Odlišnost bezpečnosti LoRaWAN ve verzi 1.1

Bezpečnost LoRaWAN ve verzích 1.0 a 1.1 se hodně liší, viz obr. 2.5. Pro odhalování a zabránění útokům typu join replay a downlink ACK byl značně vylepšen mechanizmus pro práci s čítači rámců. V nové verzi čítače rámců nemohou být opakovaně použité se stejnými klíči a jejich délka se zvýšila z 16 na 32 bit. V nové verzi pro každou downlink zprávu síťový server bude inkrementovat čítač, také již není používán pouze jeden čítač, jejich počet se zvýšil na dva. Jeden je zodpovědný za zprávy na síťové a druhý za zprávy na aplikační vrstvě. Celá architektura sítě se výrazně změnila, k aplikačnímu a síťovému serveru se přidal Join Server a také byly změněny entity potřebné pro navázání spojení. AppEUI byl nahrazen JoinEUI, ten hledá Join Server přes DNS (Domain Name System). Struktura join paketů obsahuje JoinEUI, DevEUI (8 bajt každý) a DevNonce (2 bajty). DevNonce byl změněn na hodnotu čítače (ve verzi 1.0 byla použita náhodná hodnota).

Dále byl vylepšen mechanizmus pro práci s bezpečnostními klíči, nyní jsou z NwkKey (Network Key) odvozené 3 další klíče: Forwarding Network session integrity key (FNwkSIntKey), Serving Network session integrity key (SNwkSIntKey) a Network session encryption key (NwkSEncKey). Klíče jsou zodpovědné za kontrolu integrity zpráv mezi koncovým zařízením a síťovým serverem. Na jejich vytvoření jsou použité tři entity: JoinEui, DevNonce a JoinNonce.

1. FNwkSIntKey: 128bitový klíč relace, používá se na výpočet MIC pro uplink zpráv. Síťový server zkontroluje MIC po přijetí zprávy od koncového zařízení. Derivace klíčů:

$$FNwkSIntKey = aes128\_encrypt(NwkKey, 0x01|JoinNonce|JoinEUI \\ |DevNonce|pad16)$$

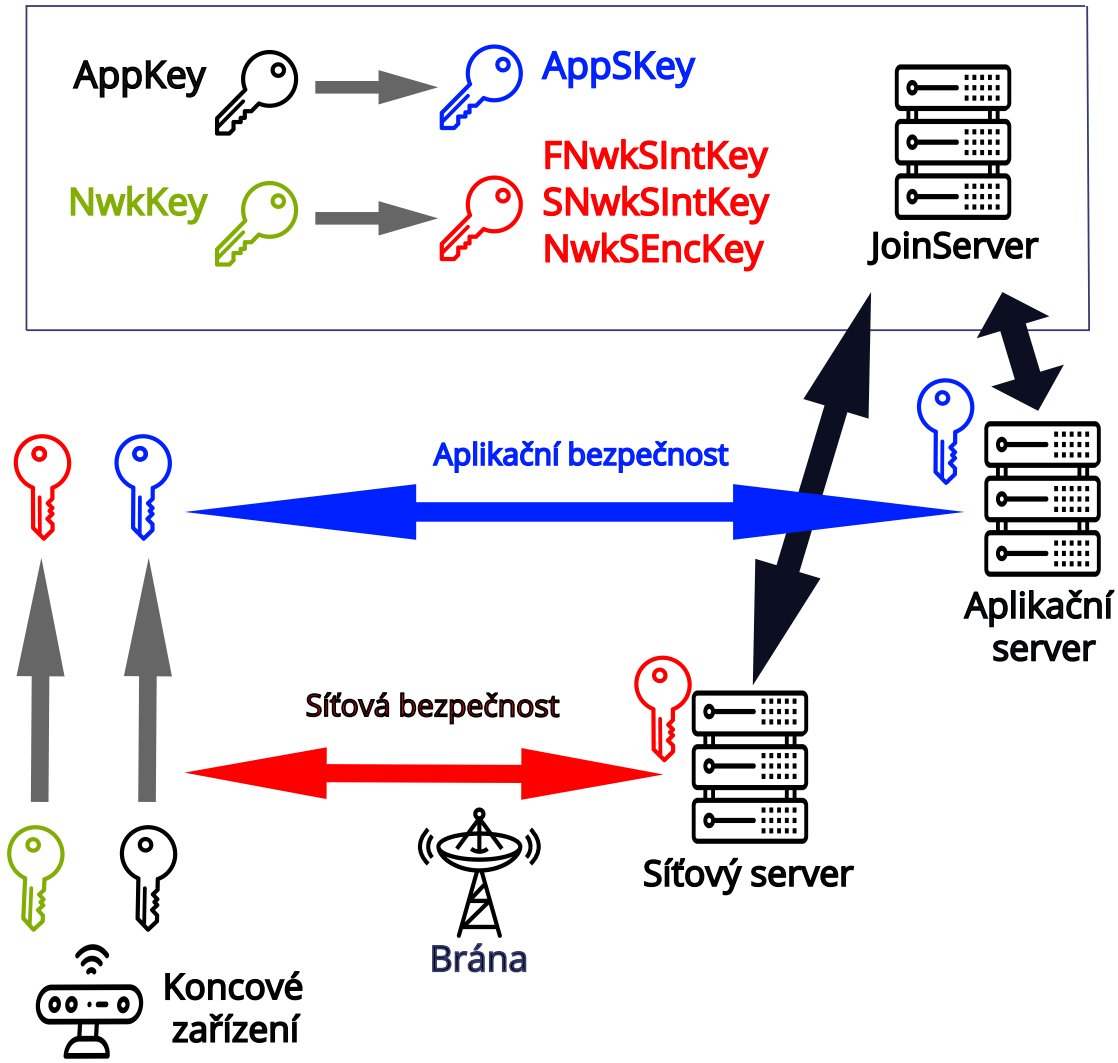
2. SNwkSIntKey: 128bitový klíč relace, používá se na vytvoření MIC pro uplink a downlink zpráv. Kontrola MIC při downlinku probíhá na koncovém zařízení. Derivace klíčů:

$$SNwkSIntKey = aes128\_encrypt(NwkKey, 0x03|JoinNonce|JoinEUI \\ |DevNonce|pad16)$$

3. NwkSEncKey: síťový klíč relace se používá na šifrování a dešifrování MAC pro uplink a downlink zpráv. Tato data obsahují pouze koncové zařízení a síťový server.
   Derivace klíčů:

$$NwkSEncKey = aes128\_encrypted(NwkKey, 0x04|JoinNonce|JoinEUI$$
$$|DevNonce|pad16)$$

Join Server provádí OTA aktivace a je zodpovědný za odvození klíčů relace. Následně je odesílá jejich síťovému a aplikačnímu serveru. Signály od koncových zařízení jsou zkoumány Join serverem pomocí JoinEUI políčka v žádosti o připojení. App a Network klíče jsou uložené pouze na Join serveru a na koncových zařízeních. Síťový a aplikační server stahují výhradně klíče relace a nikdy neznají root (AppKey, NwkKey) klíče. Několik join serverů může být připojeno do síťového serveru najednou.



Obr. 2.5: Bezpečnostní architektura verze 1.1.

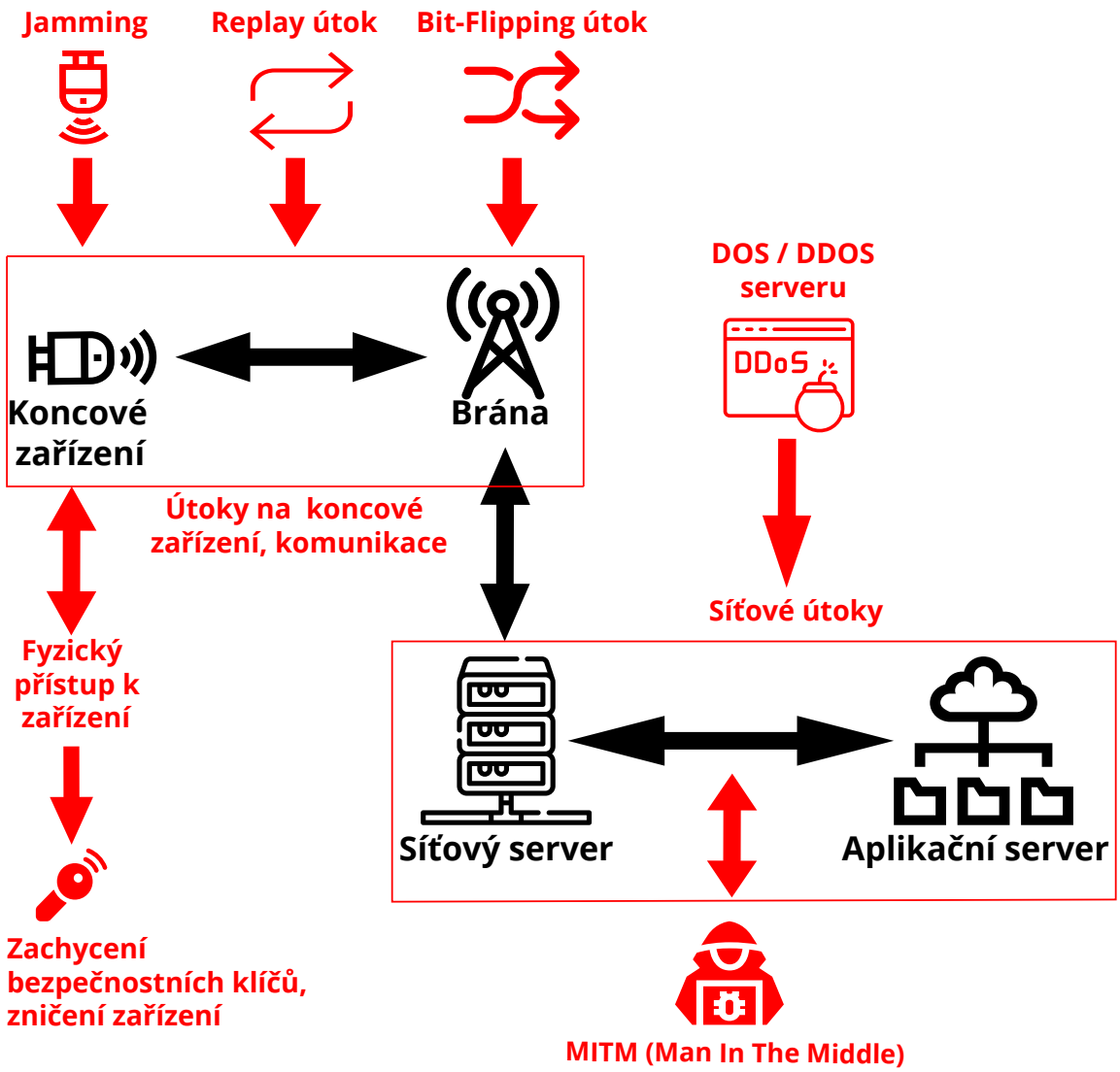
Shrnutí:

Verze 1.1 získala mnoho vylepšení oproti staré verzi. Má obnovený systém distribuce síťových klíčů relace a vylepšený mechanizmus čítače rámců, což mnohem zmenšuje rizika útoku typu join-replay. Díky symetrickému šifrování oba způsoby aktivace poskytují vysokou úroveň zabezpečení během komunikace mezi koncovým uzlem a serverem.

3 Analýza anomálií a návrh scénářů detekce

3.1 Bezpečnostní incidenty v síti LoRaWAN

Na obrázku 3.1 jsou znázorněné nejpoužívanější typy útoků na síť LoRaWAN, včetně: jammingu (rušení komunikace záplavou neověřených zpráv s vyšší hodnotou RSSI), zachycení bezpečnostních klíčů nebo zničení koncového zařízení formou fyzického kontaktu s ním, také existují útoky typu Replay (opakované škodlivé přenosy) a Bit-Flipping (přenosy s podmíněným obsahem). Kromě zmíněných se rozlišují různé typy síťových útoků, jsou to: DOS nebo DDOS serveru, MITM (Man in the middle).



Obr. 3.1: Nejčastější typy útoků na LoRaWAN síť.

3.2 Analýza incidentů

Jak bylo zmíněno výše, útoky je možné rozdělit do dvou skupin. Síťové útoky – útoky (útoky na síťový nebo aplikační server) a útoky přímo na koncové zařízení nebo na komunikace mezi zařízením a bránou.

3.2.1 Síťové útoky

Man in the middle Zatímco jsou zprávy LoRaWAN šifrovány a používají kontrolu integrity zpráv (Message Integrity Code), jejich kontrola se neprovádí ve stejných místech. Data podepsána MIC jsou kontrolována síťovým serverem, ale data podepsána AppSKey jsou ověřována již aplikačním serverem. To znamená, že mezi síťovým a aplikačním serverem nelze zkontrolovat přesnou integritu a autenticitu obsahu. V tomto bodě lze provést útok typu MITM a nějakým způsobem změnit nebo odposlouchávat komunikace.

DoS a DDoS útoky Útok DoS (Denial of Service) spočívá ve vyřazení zařízení z provozu. Existuje ve 2 variacích:

1) Obrovská záplava paketů jedním uzlem na server způsobí jeho zpomalení a zastavení z důvodu nemožnosti serveru zpracovávat takové množství. Nejznámější typy jsou: buffer overflow, ICMP flood a SYN flood.

2) Jednoduše zneužívají zranitelnosti, které způsobí selhání cílového systému nebo služby. Při těchto útocích je posílána škodlivá zpráva, která následně rozbije a odepře přístup k celému systému.

K útoku DDoS (Distributed Denial of Service) dochází, když více uzlů organizuje synchronizovaný útok DoS na jediný cíl. Zásadní rozdíl spočívá v tom, že místo útoku z jednoho místa je cíl napaden z mnoha míst najednou. DDoS poskytuje útočníkovi několik výhod:

- Místo útoků je obtížné odhalit kvůli náhodné distribuci útočných systémů (často po celém světě).

- Mají násobně větší výkon než DoS útoky.

3.2.2 Útoky na koncová zařízení

Fyzický přístup k zařízení LoRaWAN poskytuje end-to-end zabezpečení využitím aplikačních a síťových klíčů. Útočník s fyzickým přístupem však může ohrozit koncová zařízení LoRa. Pokud útočník získá fyzický přístup k zařízení, může získat klíče. Zprávy mezi hostitelem a rádiovým modulem lze zachytit pomocí externího

hardwaru. V dnešní době rádiové moduly na trhu neposkytují zabezpečení interakcí mezi MCU (Micro Controller Unit) a rádiovým modulem. Kvůli tomu nelze žádným způsobem zjistit, zda příkazy odeslané do rádiového modulu byly odeslány hostitelskou MCU, nebo jinou entitou. Útočník by také mohl zachytit všechny výměny dat mezi hostitelskou MCU a rádiovým modulem a použít zachycené informace k vytvoření škodlivého zařízení se stejnými pověřeními nebo získat přístup k užitečným datům.

Kromě toho, pokud útočník bude mít fyzický přístup k zařízení, může zařízení zničit nebo odpojit, tím zastaví vysílání dalších dat.

Rušení signálu (Jamming) Rádiové rušení je jedním z hlavních problémů při nasazení IoT [27]. Škodlivé entity mohou přenášet silný rádiový signál v blízkosti aplikačních zařízení, a tak rušit rádiové přenosy. Přenosy se stejnou frekvencí a stejným rozptylovým faktorem se mohou navzájem rušit [28]. Tato bezpečnostní slabina ve fyzické vrstvě LoRa umožňuje nebezpečným entitám nebo třetím stranám používat zařízení typu COTS (commercial-off-the-shelf) k zasekávání sítí.

Tyto útoky obvykle vyžadují speciální hardware, např. platformu Arduino s modulem LoRa. Kdokoli může zaplavit určitou frekvenci, aby přerušil všechny přenosy. Jsou však řešení, například rušení celé sítě může být snadno detekováno. Všechna zařízení, která komunikují na stejné frekvenci, totiž začnou ze sítě najednou vypadávat. Správci sítě tomu mohou lehce zabránit, jednoduše přepnou provozní frekvence. Rádiové rušení je jedním z nebezpečných problémů pro IoT sféru.

Existují různé typy jammingu:

1. **Triggered Jamming** – aby se zabránilo několika současným přenosům, mají rádiové moduly LoRa schopnost skenovat určitý kanál na přítomnost probíhajícího přenosu. Útočníci mohou tuto možnost zneužít k detekci aktivity na konkrétním kanálu. Jakmile je na kanálu detekován přenos, škodlivé zařízení LoRa může začít vysílat, aby se tento přenos zasekl. Spoléhá se většinou na detekci preambulových symbolů a poté na rušení zařízení bez demodulace nebo dekódování jakékoli jiné části signálu. Tento útok se zvláště hodí k rušení sady vysílajících bran, protože ruší bez rozdílu všechny přenosy.
2. **Selective Jamming** – je pravděpodobně nejefektivnější technikou rušení [29]. Oproti triggered jammingu má řadu výhod, například není tak snadno detekovatelný, protože neovlivňuje všechna zařízení na konkrétní frekvenci. Zasekává pouze vybraná zařízení nebo zprávy, a proto nelze přesně detekovat, jestli někdo provádí útok nebo došlo k jiným technickým problémům. Selektivní rušení vyžaduje, aby byla určitá část zprávy přijata a přečtena před provedením akce. Pro tuto techniku jsou nezbytné přísně požadavky na dodržení času, zbývající doba pro úspěšné zaseknutí paketu je tedy při selektivním rušení kratší.

Replay útoky Během OTAA aktivaci koncového zařízení útočník může za-

chytit aktivační zprávu a následně také požádat o spojení se serverem duplikovanou zprávou. Žádost by měla obsahovat stejné hodnoty AppEUI, DevEUI a DevNonce jako původní zařízení. Když server obdrží takový požadavek, začne jej porovnávat s dříve uloženými informacemi. V případě, kdy server obdrží zprávu s předchozí hodnotou DevNonce, ve většině případů takovou zprávu zahodí. Servery mohou používat náhodnou hodnotu DevNonce, nebo inkrementovat ji po každém přenosu. Jestliže se používá varianta s inkrementací, tak je server je velmi náchylný na replay útoky kvůli tomu, že zachycením aktivační zprávy lze přesně stanovit další očekávanou hodnotu DevNonce a poté takovou zprávu odeslat před skutečným zařízením.

Bit-flipping útok Útoky typu bit-flipping jsou schopné měnit zašifrovaná pole textu bez jejich dešifrování. Změna může způsobit poškození velkého počtu dat během vysílání. Bit-flipping útok lze provést ve specifických šifrovacích režimech, kde má text stejné bitové pořadí jako šifrovaný text [30]. V LoRaWAN sítích zařízení používají blokový režim šifrování – CTR (Counter) [31], ale text šifrovaný v CTR módu je náchylný na Bit-Flipping útoky [32]. Proces funguje tím způsobem, že režim CTR používá na šifrování XOR, ale ten během šifrování nemění pořadí bitů. Stejné pořadí bitů dovoluje provést útok.

Výpadek brány Může nastát, pokud někdo zaútočí na kabely brány a odpojí je od sítí. Brána následně nebude schopna odeslat data na síťový server. V případě použití Wi-Fi propojení se serverem existuje možnost zrušit internetové propojení například použitím Wi-Fi jammeru. Ten pošle škodlivý rámec na směrovač s příkazem na odpojení od jména připojených a fungujících zařízení v této síti, tím všechna zařízení v síti odpojí (denial of service).

3.3 Možnosti detekce

3.3.1 Detekce síťových útoků

Detekce útoku typu MITM spočívá v analýze síťového provozu, například při zdlouhavých výpočtech hašovacích funkcí. Strany mohou kontrolovat nesrovnalosti v době odezvy. Předpokládejme, že dvě strany obvykle stráví určitý čas na dokončení konkrétní transakce. Pokud však jedna transakce vyžaduje delší dobu na odeslání, může to znamenat zásah třetí strany.

Detekce útoku typu DOS nebo DDOS je většinou rozeznatelná bez žádných nástrojů nebo technik Existují však způsoby, jak útok detekovat ještě před tím, než systém spadne nebo začne brzdit. Jedním z těchto způsobů je použití systému, který bude nastavovat konkrétní limit a kontrolovat určitý počet paketů a jejich typ ke konkrétní IP adrese. Po překročení limitu bude systém dávat signál o možném útoku.

3.3.2 Detekce útoků na koncová zařízení

Detekce fyzického zasahování do zařízení spočívá v monitorování pravidelného času vysílání v reálném čase a přítomnosti provozu na bráně nebo serveru. Když data přestanou přicházet, znamená to, že zařízení nefunguje a možná se jedná o útok nebo nějakou jinou událost.

Detekce rušení signálu lze poznat podle pravidelného kontrolování logů na bráně nebo serveru. Jammer většinou bude mít větší hodnotu RSSI (Received Signal Strength Indicator), SNR (Signal-to-noise ratio) a častý interval vysílání. Porovnáním hodnot během vysílání je možné stanovit, v jaký časový okamžik se jedná o útok nebo nějakou anomálii.

Detekce útoku typu Replay lze provádět pomocí hodnoty DevNonce. Při aktivaci koncového zařízení jsou tyto hodnoty nastaveny na 0 a každá zpráva přicházející z brány nebo zařízení bude inkrementovat hodnotu. Server po každé zprávě bude porovnávat novou hodnotu DevNonce s těmi, které už přišly. Pokud přijde zastaralá hodnota DevNonce, tak takovou zprávu odmítne. Lepším způsobem je však použití náhodných hodnot DevNonce. Použitím náhodných hodnot DevNonce lze snížit pravděpodobnost opakovaného vysílání zprávy škodlivou entitou, protože útočník nebude schopen spočítat další správnou hodnotu po zachycení zprávy.

Detekce útoku typu Bit-Flipping probíhá stejným způsobem jako detekce útoků typu Replay. Útočník nebude schopen odeslat škodlivou zprávu, v případě, kdy server bude pravidelně kontrolovat hodnoty DevNonce a odmítat zprávy se špatným DevNonce.

Detekce výpadku brány stejně jako při rušení signálu probíhá většinou pomocí pravidelného kontrolování logů na bráně nebo příchozího provozu na serveru.

3.4 Návrh scénářů detekce incidentů

Jak už bylo zmíněno v předchozí kapitole, detekce rušení signálu a výpadku brány obvykle probíhá cestou pravidelného monitorování logů na bráně nebo na serveru.

Tímto postupem lze zabránit různým útokům, například fyzickým, kdy se útočník dostane k zařízení a vypne ho nebo zničí. Pomocí analýzy příchozích dat na bránu si lze jednoduše zjistit výpadek sítě nebo fyzické poškození koncového zařízení.

3.4.1 Scénáře detekce jammingu

Detekce na bráně

Detekce by se měla vykonávat pomocí pravidelného monitorování logů na bráně. Pro detekci lze použít „Mosquitto mqqt broker“ [33]. Monitorování provozu pomocí

Wiresharku v tomto případě nebude fungovat kvůli tomu, že Wireshark naslouchá provoz na Ethernet rozhraní. V tomto bodě již data nejsou čitelná, proto je potřeba monitorovat provoz před tím, než data budou převedena na formát vhodný pro síťový server. Mqqt broker bude naslouchat na určitém portu a jakmile data budou zachycena branou, zobrazí je.

Následně, na základě analýzy logů, bude možnost stanovit, zda se jednalo o bezpečnou událost na základě různých hodnot RSSI. Hodnoty RSSI totiž v případě jammingu bývají vyšší než u koncového zařízení nebo velmi častého vysílání.

Detekce na serveru

Jiná možnost je použití serveru na detekci. Pro zkoumání provozu na serveru lze použít API nebo grafické rozhraní serveru. Výhodou oproti bráně je to, že detekce prochází v reálném čase. Tímto způsobem lze momentálně detekovat a zabránit útokům nebo jiným bezpečnostním událostem.

API serveru dovoluje vytvoření vlastního nástroje na detekci. Nástroj by měl fungovat v reálném čase a obsahovat grafické rozhraní. Detekce útoků by probíhala na základě střední hodnoty RSSI nebo průměrného času vysílání koncového zařízení. Nástroj by měl rozlišovat normální stav vysílání od stavu útoků a také upozorňovat na škodlivé zprávy. Ideálním řešením by bylo implementovat mapu a poté na ní ukazovat polohu vysílajícího zařízení. Popsaným způsobem je možné stanovit přesné místo vysílání škodlivého zařízení a lze co nejrychleji tento jammer zneškodnit.

3.4.2 Scénáře detekce výpadku brány

Detekce na bráně

Při detekci výpadku brány ze sítě je potřeba kontrolovat připojení brány na server. Pro tento účel lze použít nástroj T-shark, jedná se o terminálovou verzi Wiresharku. T-shark zapisuje zachycený provoz do .pcap souborů, což je formát používaný Wiresharkem a různými dalšími nástroji na sledování síťového provozu. Zkoumání zachycených dat na bráně bez grafického rozhraní není zcela pohodlný způsob. Vhodným řešením je přenést .pcap soubory na počítač pro další testování v GUI (Graphical User Interface) prostředí Wiresharku, které poskytuje detailnější analýzu dat než T-shark. Existuje několik možností jak přenést soubor z brány na počítač, například SCP (Secure Copy Protokol), SFTP (SSH File Transfer Protocol) klient, USB (Universal Serial Bus) driver nebo použitím různých servisů pro vzdálený management dat.

V zachycených datech při odpojení je možné vidět příznaky pokusů o opětovné navázání spojení se serverem. Podle toho lze poznat, že se brána odpojila nebo porouchala.

Detekce na serveru

Při použití serverů na detekci je potřeba kontrolovat příchozí provoz od koncového zařízení. Když data přestanou přicházet na server, znamená to, že se brána nebo koncové zařízení porouchalo.

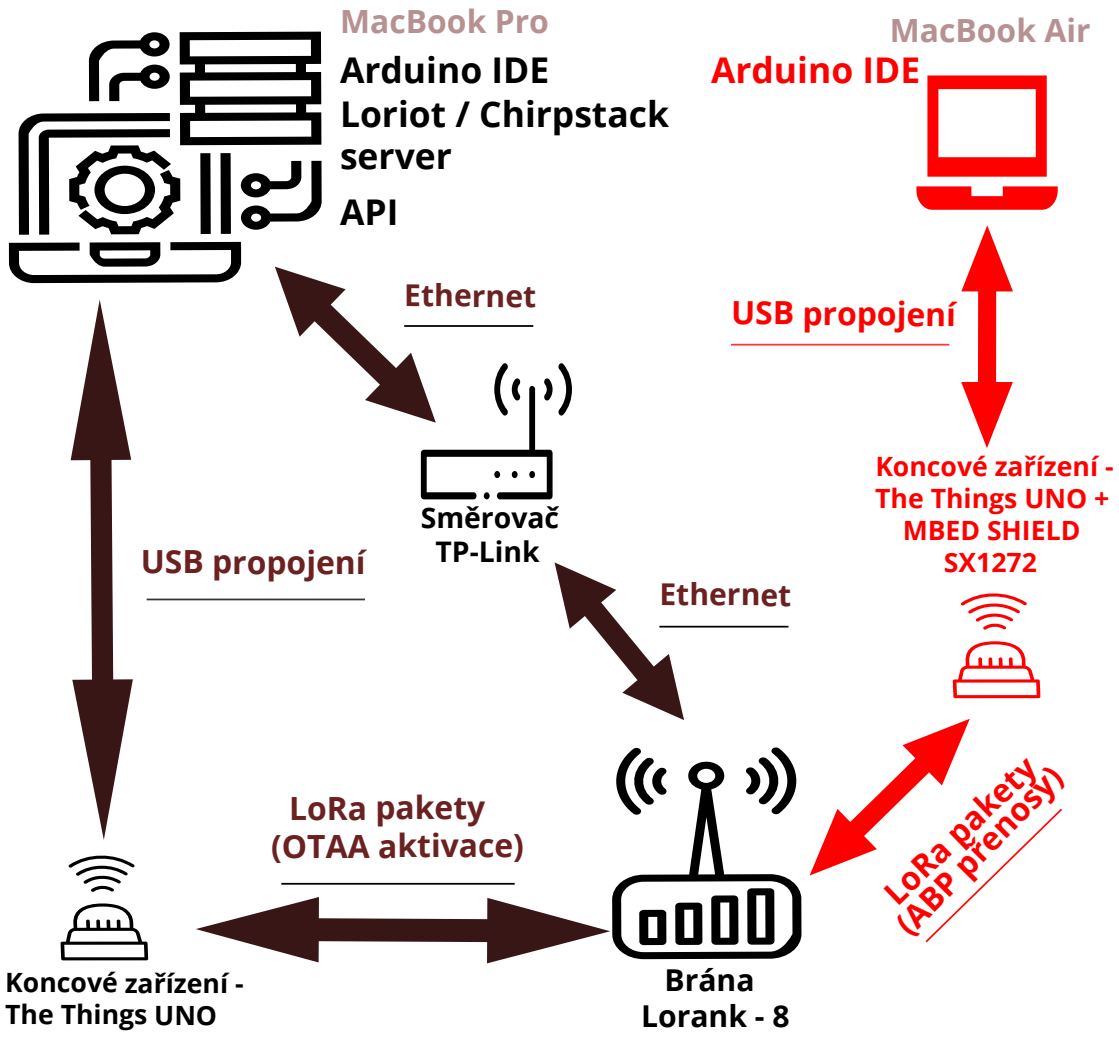
Použitím API serveru lze implementovat detekční systém výpadku brány. Fungovat by mělo tím způsobem, že systém pravidelně kontroluje příchozí provoz. Jestliže data nebudou přicházet delší dobu, nastane kontrola, zda se odpojila právě brána (aby se nejednalo o výpadek koncového zařízení) a následně stav oznámí.

4 Realizace navržených scénářů detekce

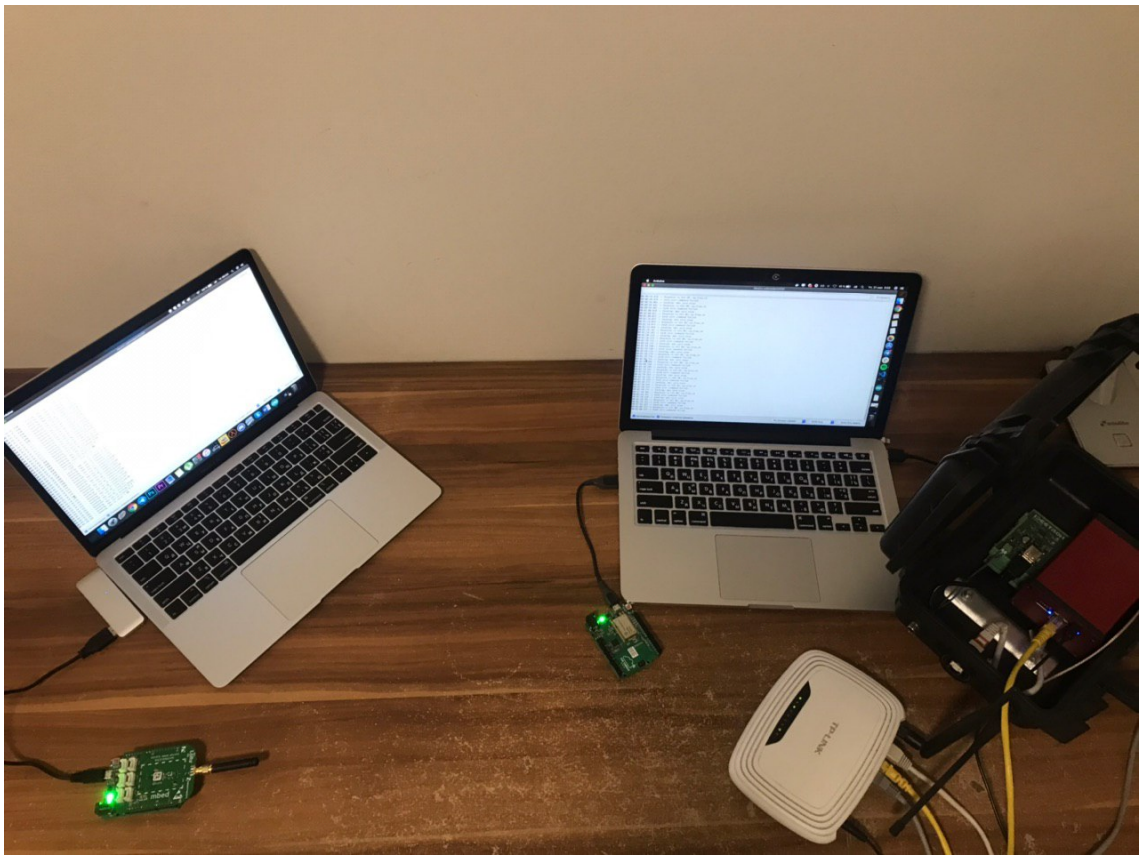
Pro lepší pochopení architektury sítě a v dnešní době existujících nástrojů byly zprovozněny dvě varianty sítě s různými síťovými servery. Jako první varianta byla použita LoRaWAN síť s hotovou a nakonfigurovanou infrastrukturou – Loriot Cloud. Druhou variantou byl server Chirpstack, který potřeboval konfigurace.

4.1 Testovací prostředí

Pro testovací účely byla zprovozněna vlastní LoRaWAN síť ve verzi 1.0 (kvůli dostupným prostředkům) a to znázorněným způsobem: obr. 4.1 a 4.2.



Obr. 4.1: Schéma zapojení sítě a použité entity.



Obr. 4.2: Zapojení sítě a použité entity.

4.1.1 Prvky sítě

V rámci testováni byly použity další entity:

1. 2x Počítač – MacBook Air a MacBook Pro.
2. 2x Arduino – The Things UNO.
3. 2x USB kabel.
4. 3x Ethernet kabel.
5. 1x SX1272 Mbed Component Shield.
6. 1x Směrovač TP-LINK.
7. 1x Brána Lorank-8.

Koncové zařízení

Jako koncové zařízení bylo vybráno Arduino – The Things Uno [34]. Jedná se v podstatě o obyčejné Arduino Leonardo s LoRaWAN modulem a vlastní anténou. Zařízení lze lehce využít se stávajícím IDE (Integrated Development Environment) od Arduino. Pracuje na frekvenci 868 MHz, což je vyhovujícím pro EU. The Things UNO je kompatibilní s různými shieldy a nepotřebuje pro svou činnost žádnou kon-

figuraci. Je ideálním řešením pro návrh privátní sítě a testování různých návrhů a konceptů v IoT.

Jammer

Jako jammer bylo zvoleno koncové zařízení Thé Things Uno v kombinaci s Mbed shieldem SX1272 [35]. Jedná se o vysílač vybavený modemem s dlouhým dosahem – LoRaTM. Ten poskytuje vysokou odolnost proti rušení a minimalizaci spotřeby proudu. Vysoká citlivost v kombinaci s integrovaným výkonovým zesilovačem +20 dBm poskytuje špičkový rozpočet na propojení (link budget), díky čemuž je optimální pro každou aplikaci vyžadující rozsah nebo robustnost a dovoluje následně zvýšit RSSI (Received Signal Strength Indicator) hodnoty během vysílání. Tímto způsobem následně ruší přenosy jiných zařízení vysílajících bez shieldu.

Brána

Lorank-8 se skládá z řídícího modulu Beaglebone Black. Hlavním úkolem tohoto modulu je přesměrování veškerých LoRa rámců na kabelové ethernet spojení nebo 4g pakety, a naopak v případě downlink zpráv. Bránu vyrábí firma Ideetron. Dovoluje řídit osm spojení najednou [36]. Na bráně je použit operační systém Linux (verze 7 Wheezy), aby bylo možno bránu konfigurovat, lze se k ní připojit pomocí SSH (Secure Shell).

Loriot Cloud

LORIOT je švýcarská společnost v oblasti internetu věcí. Poskytuje vhodný cloud pro návrhy v IoT a také RESTful API, které dovoluje poměrně snadné přenášení a zpracování dat na zařízení pomocí protokolu HTTP, MQTT a Web-Socketu.

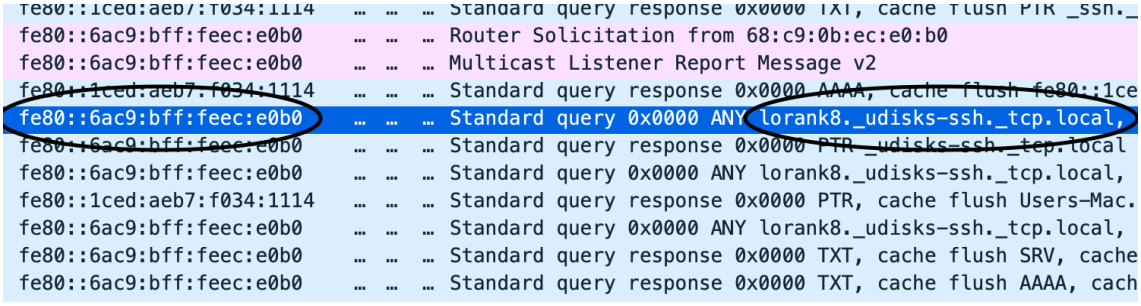
Chirpstack server

Chirpstack server poskytuje komponenty s otevřeným zdrojovým kódem pro sítě LoRaWAN [37]. Dříve byl známý jako LoRa server, následně byl přejmenován na ChirpStack. Je to řešení, které má modulární strukturu, a díky tomu se jednoduše integruje do stávajících infrastruktur. Podporuje tři druhy zařízení: A, B a C, různé kanály: EU, US a AU, sledování provozu v reálném čase, verzi LoRaWAN 1.0, 1.1 a dovoluje vytváření několika aplikací. Poskytuje webové rozhraní pro správu aplikací, bran a koncových zařízení. Taktéž má svoje REST a gRPC API, což dále dovoluje použití dat ze serveru na externích zdrojích.

4.1.2 Zprovoznění LoRaWAN sítě a serveru

Síť funguje následujícím způsobem: počítač je propojen s koncovým zařízením pomocí USB kabelu a vysílá LoRa pakety na bránu. Vysílání probíhá pomocí programu Arduino IDE [38] a knihovny The Things Network [39].

Brána zachycuje příchozí pakety a pomocí packet forwarderu a ethernetu je přeposílá na síťový server přes směrovač, ten umožnil zapojit počítač a bránu do jedné podsítě. Aby bylo možné konfigurovat bránu, bylo nutné zjistit její IP adresu, proto byl použit program Wireshark. Je to vhodný nástroj na skenovaní sítí a zachycení paketů. Brána požaduje IP adresu z DHCP serveru, poté lze zachytit příchozí pakety pomocí Wiresharku a zjistit její IP adresu, viz obr. 4.3. Celá konfigurace brány probíhala pomocí SSH.



Obr. 4.3: IPv6 adresa brány ve Wiresharku.

Síťový server dokáže odstranit duplicity v příchozích paketech, dešifruje zprávy a přepošle je na aplikační server. Aplikační server obsahuje API, se kterým je možné dále jednoduše pracovat a zpracovávat příchozí provoz.

Konfigurace a použití serveru Loriot

Nejdříve bylo nutné propojit prvky sítě mezi sebou. Počítač s koncovým zařízením přes USB, poté počítač a bránu se směrovačem pomocí ethernet kabelu.

Brána byla zaregistrována na cloudu Loriot. Z cloudu byl stáhnout software, který dovolil napojit bránu na server. Příkazy pro stahování nejnovější konfigurace a přihlašování na server Loriot, viz výpis 4.1:

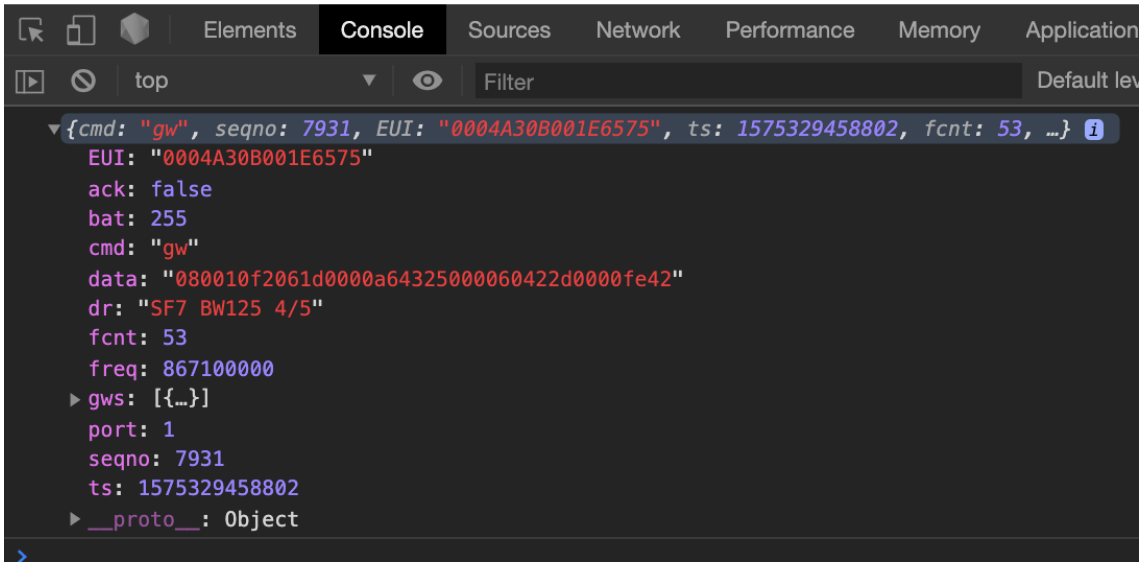
Výpis 4.1: Přihlašování na server Loriot.

```
1 cd /tmp #otevrit slozku tmp
2 wget http://www.loriot.io/home/gwsw/loriot-lorank-
3    8-ic880a-SPI-0-latest.sh #stahnout script
```

```
#pridat pravo na spousteni
chmod +x loriot-lorank-8-ic880a-SPI-0-latest.sh
#spustit skript
./loriot -lorank -8-ic880a -SPI -0-latest.sh -n
```

Jakmile byla brána připojena na síťový server, bylo možné na serveru vytvořit vlastní aplikaci a začít vysílat z koncového zařízení na cloud. K aplikaci byl připojen koncový prvek, v rámci této práce – The Things Uno, ze kterého byly posílány zprávy. Brána zachytila LoRa rámce, Beaglebone modul rámce zpracoval a pomocí ethernetu je odeslal na cloud. Ten je přijal, odstranil duplicity a zpracoval.

Stahování dat probíhalo pomocí zabudovaného API v cloudu a protokolu WebSocket. Zmíněný protokol je vhodnou alternativou HTTP (HyperText Transfer Protocol) spojení, jeho výhodou je real-time komunikace klienta se serverem. Příchozí odpověď je v json formátu, který je možné snadno rozparsovat a dále pracovat jako s obyčejným objektem, viz obr. 4.4.



Obr. 4.4: Výsledky přenosu.

Konfigurace a použití serveru Chirpstack

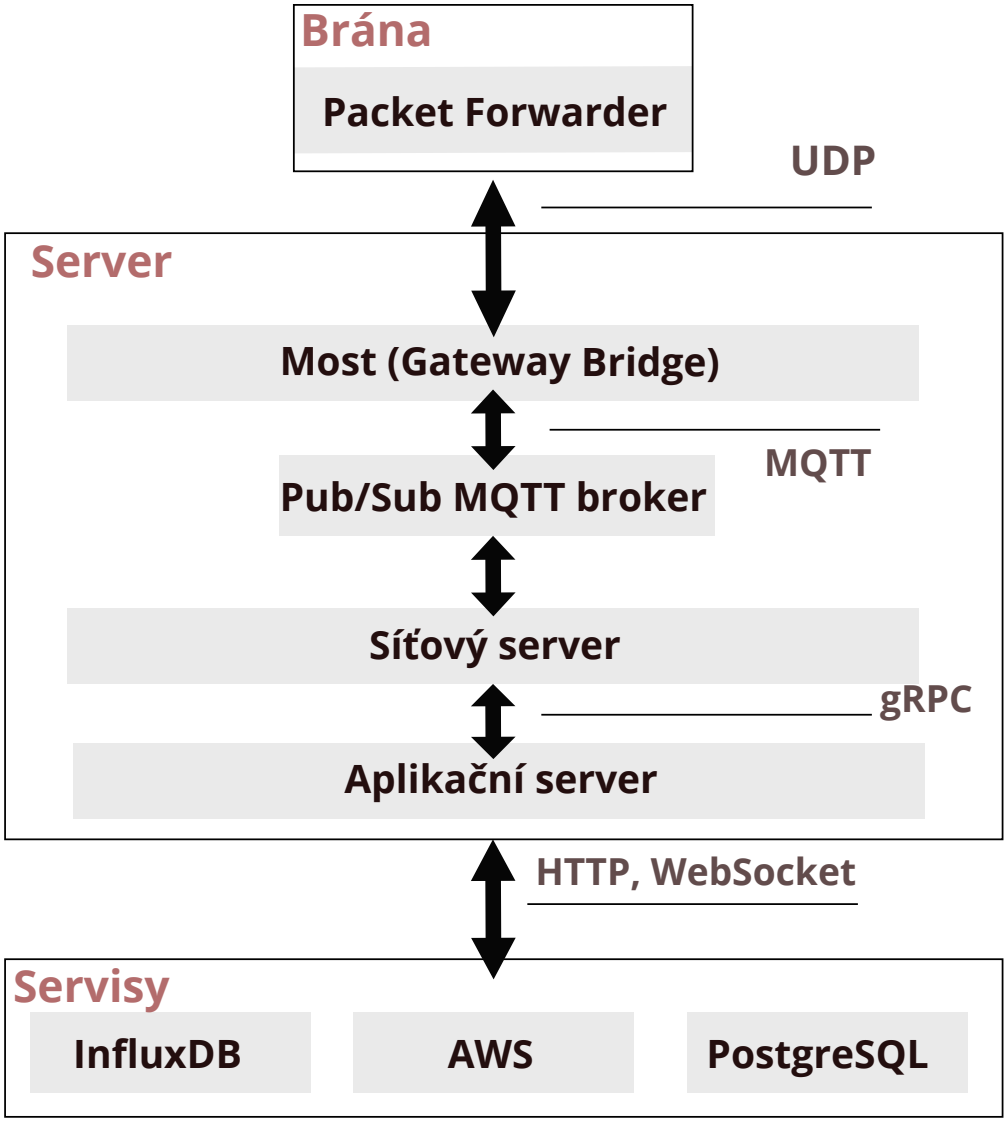
Postup zapojení je zcela totožný, rozdíl je především v konfiguraci síťového serveru a nastavení brány.

Po zapojení všech prvků sítě bylo potřeba nastavit packet forwarder na bráně. Je to program, který přijímá příchozí pakety a poté je posílá na síťový server. V konfiguračním souboru packet forwarderu se nastavoval port serveru a IP počítače, na

kterém běžel server. Na konfiguraci vlastního serveru byl použit docker – pro další pochopení je docker detailněji popsán.

Docker – jeden z druhů virtualizace. Jedná se o nástroj, který umožňuje zabalit aplikace ve standardizované spustitelné komponenty – kontejnery, které kombinují zdrojový kód aplikace se všemi knihovnami operačního systému a závislostmi vyžadovanými ke spuštění kódu do standardizovaného modulu. Na rozdíl od virtuálních počítačů kontejnery nezatěžují systémy do takové míry, takže je možné efektivně využívat prostředky systému.

Server se skládá s různých částí: síťový most, síťový server, aplikační server.



Obr. 4.5: Architektura Chirpstack serveru.

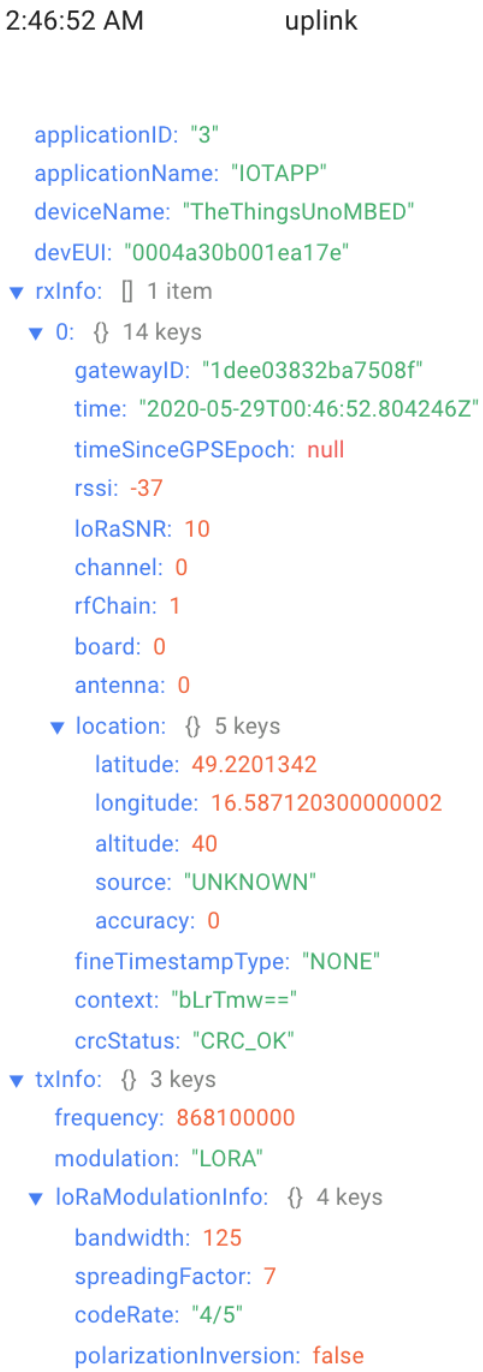
Všechny části běží v různých kontejnerech pomocí nástroje – docker-compose, ten dovoluje spustit všechny kontejnery společně.

Servery budou komunikovat, zpracovávat a vyměňovat si data mezi sebou. Proces funguje následujícím způsobem: koncové zařízení odesílá pakety, brána je zachytí a pomocí nakonfigurovaného packet forwarderu dále přepošle na adresu a port síťového mostu. Most bude přijímat na určité adrese a očekávat pakety zaslané bránou. Po přijetí paketů je odešle do MQTT zprostředkovatele. Jeho cílem je převést pakety na formát vhodný pro síťový server, může to být protobuf nebo json.

Síťový server je zapsán na MQTT zprostředkovateli, a tak získává data v případě jeho obnovení. Hlavními funkcemi jsou autentifikace, zpracování dat při downlinku, odstranění duplicit v příchozích paketech a další komunikace s aplikačním serverem.

Aplikační server je zodpovědný za zpracování žádostí o připojení, šifrování užitečných obsahů aplikace, navíc nabízí webové rozhraní. S jeho pomoci lze spravovat uživatele, organizace, aplikace a zařízení. Pro integraci s externími službami nabízí rozhraní RESTful (Representational State Transfer) a gRPC (general-purpose Remote Procedure Calls) API. Po úspěšné ABP aktivace zařízení na serveru je vidět uplink zprávu, viz obr. 4.6. Při OTAA aktivace je vidět Join Request na serveru viz obr. 4.7.

Data zařízení lze odesílat nebo přijímat přes MQTT, HTTP a zapisovat do různých databází.



Obr. 4.6: Přijatá zpráva po úspěšné ABP aktivaci zařízení na serveru Chirpstack.

Obr. 4.7: Přijaté Join Request během OTAA aktivace zařízení na serveru Chirpstack.

4.2 Simulace bezpečnostních incidentů

V rámci bakalářské práce byly nasimulovány dva bezpečnostní incidenty: jamming (rušení vysílání koncového zařízení jiným zařízením) a výpadek brány ze sítě.

4.2.1 Jamming

Jak již bylo zmíněno, pro úspěšné navázání spojení se serverem je potřeba aktivovat koncové zařízení, tím může být OTAA nebo ABP aktivace. OTAA se považuje za bezpečnější verzi, oproti ABP během aktivace prochází distribuce klíčů. Vysílání obvykle probíhá na různých kanálech, může mít různé šířky pásma a SF. Jsou to důležité podmínky pro zajištění bezpečnosti a vyhýbání simulativním přenosům s jinými zařízení.

Pokud se používá více kanálů, často nebude jeden jammer schopen koncové zařízení ovlivnit, protože po každém odeslání dat nebo aktivačních zpráv bude koncové zařízení měnit kanál (frekvence), na kterém vysílá (např. 868.1 na 868.5 atd.). To lze vyřešit nastavením většího počtu jammerů, kdy by každý vysílal na různých kanálech, a tak zablokovat většinu defaultních kanálů. Pro úspěšný útok bylo tudíž potřeba nastavit jeden kanál vysílání na zařízení a jammeru (viz výpis 4.2). Byla použita knihovna v jazyce C – The Things Network.

Výpis 4.2: Metoda na odpojování kanálů.

```
1  void TheThingsNetwork::configureEU868()
2  {
3    // vypnuti defaultne pouzivanych kanalu 1 - 8
4    int i;
5    for(i = 1; i <= 8; i++){
6      sendChSet(MAC_CHANNEL_STATUS, i, "off");
7    }
8
9    char buf[10]; // bufer
10   uint32_t freq = 867100000; // frekvence
11   uint8_t ch = 0; // kanal
12
13   sprintf(buf, "%lu", freq); // vykresleni aktualni informace
14   sendChSet(MAC_CHANNEL_FREQ, ch, buf); // nastaveni kanalu
15   sendChSet(MAC_CHANNEL_STATUS, ch, "on"); // zapnuti kanalu
16 }
```

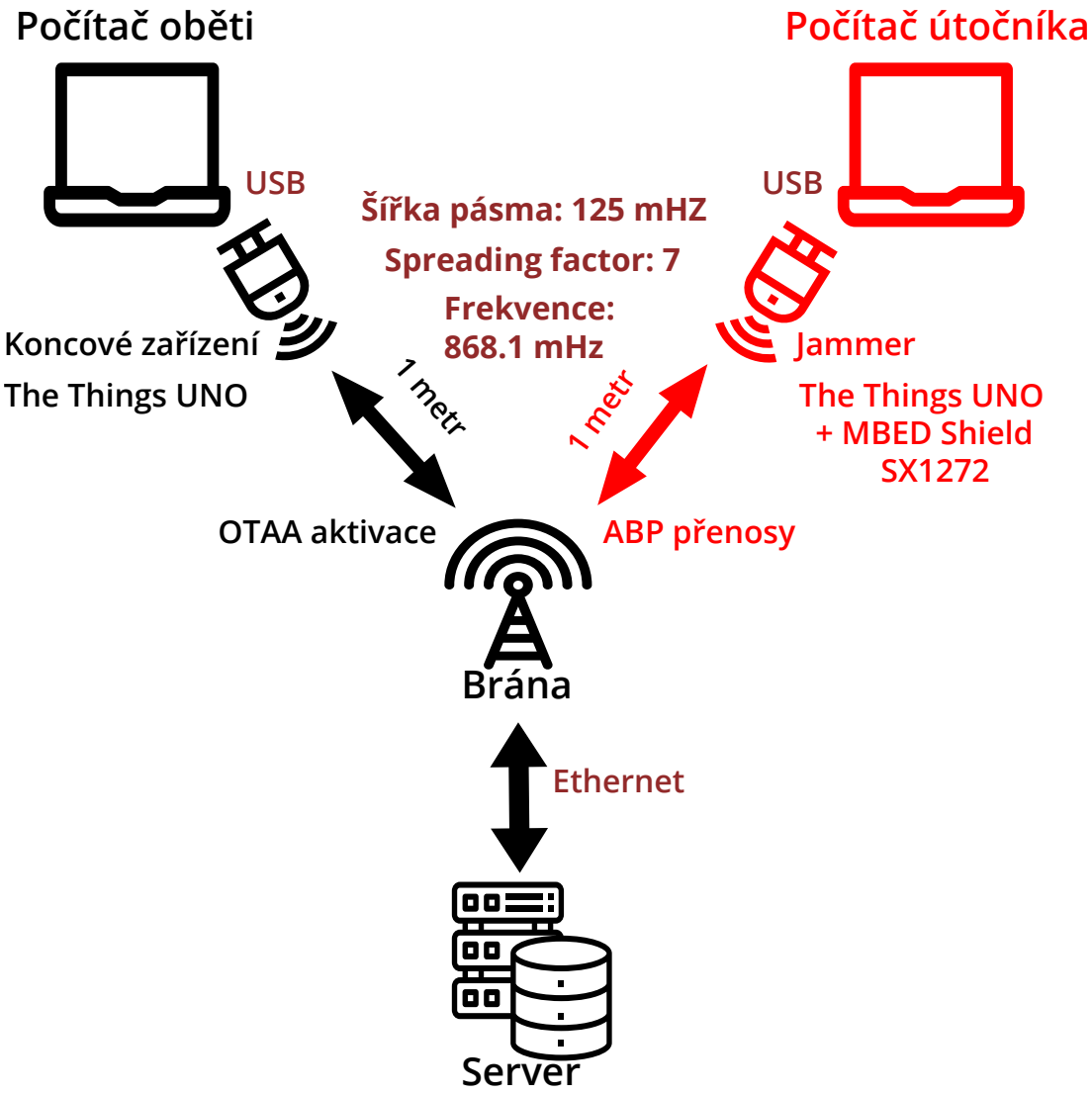
Pro fungování celé sítě byl nakonfigurován Chirpstack server. Pro činnost koncových zařízení byla použita IDE od Arduino. Jammer se nacházel ve vzdálenosti cca 1 metr od skutečného zařízení, viz obr. 4.8.

Celý proces probíhal tak, že koncové zařízení posílalo OTAA aktivační dotaz na server jednou za minutu. Jammer vysílal ABP zprávy na stejném kanálu se stejnou šířkou pásma paralelně s časovým intervalem 500 ms pak po 1 sekundě a následně po 2 sekundách. Byly použity právě ABP přenosy kvůli tomu, že před vysíláním dat nevyžadují provádět Join Requesty na rozdíl od OTAA.

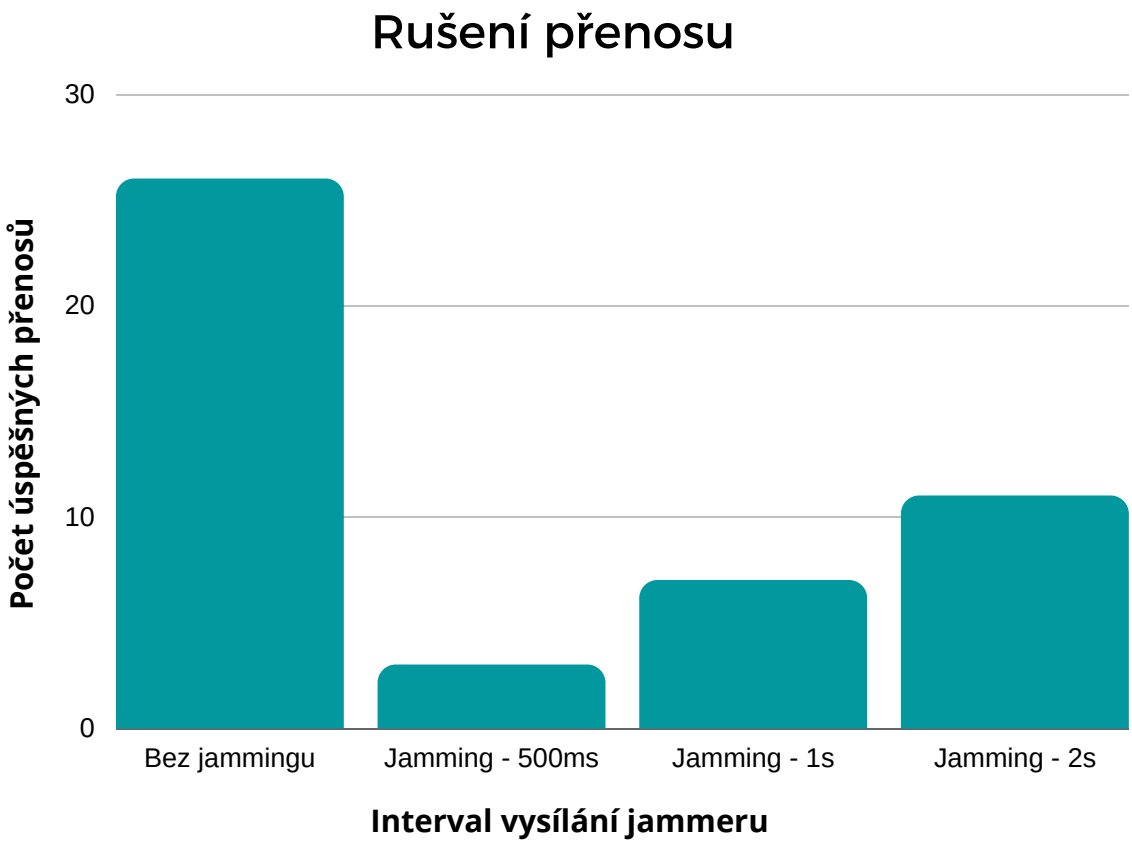
Výsledky jammingu je vidět na obr. 4.9. Po vypadnutí původního zařízení bude konzole obsahovat informace o tom, že koncové zařízení není schopné odeslat OTAA nebo data a že je kanál aktuálně obsazen, viz obr. 4.10.

Bez rušení koncové zařízení odeslalo celkově 27 aktivačních zpráv ze 30, když poté jammer začal vysílání v časovém intervalu 500 ms, koncové zařízení odeslalo pouze 3 zprávy. V intervalu 1 sekunda prošlo 7 zpráv a v intervalu 2 sekundy prošlo 9 zpráv. Tady je vidět korelace mezi časem vysílání jammeru a úspěšnou aktivací zařízení – čím rychleji se posílají data z jammeru, tím větší pravděpodobnost, že dojde k synchronnímu přenosu a následně se aktivace poruší.

Na obr. 4.8 je možné pozorovat, že se jammmer skládá z The Things Uno a Shieldu. Shield umožňuje vysílání s vyšší hodnotou SNR (Signal-to-noise ratio) a RSSI, což dovoluje efektivněji rušit přenosy. Na koncovém zařízení během vysílání je 6 SNR a RSSI je okolo -50 dBm, Shield zajišťuje vysokou odolnost proti rušení. Díky tomu je jammer schopen vysílat s hodnotou SNR 10 a RSSI okolo -35 dBm.



Obr. 4.8: Rušení přenosů koncového zařízení pomocí jammeru.



Obr. 4.9: Korelace mezi časem vysílání jammeru a počtem úspěšných přenosů.

```
21:53:46.373 -> Sending: mac join otaa
21:53:46.408 -> Response is not OK: no_free_ch
21:53:46.408 -> Send join command failed
21:53:56.403 -> Sending: mac join otaa
21:53:56.440 -> Response is not OK: no_free_ch
21:53:56.440 -> Send join command failed
21:54:06.421 -> Sending: mac join otaa
21:54:06.457 -> Response is not OK: no_free_ch
21:54:06.457 -> Send join command failed
21:54:16.465 -> Sending: mac join otaa
21:54:16.465 -> Response is not OK: no_free_ch
21:54:16.465 -> Send join command failed
21:54:26.483 -> Sending: mac join otaa
21:54:26.517 -> Response is not OK: no_free_ch
21:54:26.517 -> Send join command failed
21:54:36.510 -> Sending: mac join otaa
21:54:36.510 -> Response is not OK: no_free_ch
21:54:36.510 -> Send join command failed
```

Obr. 4.10: Obsazený kanál při odeslání.

4.2.2 Výpadek brány

Výpadek brány je poměrně častým problémem v LoRaWAN sítích. Taková situace může nastat třeba v případě, kdy nefunguje Internet a brána nebude schopna přeposlat pakety dále do síťového serveru. Také se může stát, že někdo naruší internetové propojení. To se ve většině případů týká Wi-Fi signálu, je více náchylnější na rušení než Ethernet spojení.

Dále může dojít k fyzickému porušení kabelu. Můžeme vzít do úvahy útoky na bránu v případě, kdy útočník bude znát polohu brány a zaútočí na kabely. Dalším příkladem může být náhodná porucha kvůli vnějším okolnostem (např. špatné počasí). Kromě porušení kabelu je nutné zmínit mechanické odpojení brány nebo výpadek proudu, lze rozlišit možnost útoku a náhodného odpojení.

Mezi další potíže patří i fyzické porušení brány nebo poškození systému na kterém brána funguje.

4.3 Detekce bezpečnostních incidentů

4.3.1 Detekce jammingu

Detekce jammmingu pomocí analýzu logů na bráně

Na bráně lze zkontrolovat příchozí provoz pomocí zapnutého monitoringu sítě. Podle navržených scénářů je na detekci použít mqqt broker, ten přijímá na portu 1883 data od koncového zařízení. Broker je potřeba nainstalovat na bránu a následně spustit, viz výpis: 4.3.

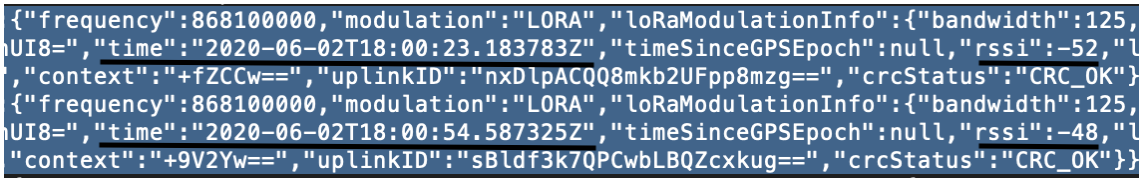
Výpis 4.3: Instalování a spuštění brokerů.

```
1  # Prikaz na obnoveni paketu Ubuntu
2  sudo apt-get update
3
4  # Prikaz na instalovani brokeru
5  sudo apt-get install mosquitto
6
7  # Prikaz na spusteni brokeru
8  # -h - host, -p - port, t - filter provozu
9  mosquitto_sub -h 192.168.1.100 -p 1883 -t gateway/
```

Po spuštění broker začne monitorovat příchozí provoz na bránu. Na obrázku 4.11 koncové zařízení vysílalo požadavky na připojení k serverů jednou za 30 sekund,

RSSI hodnoty během přenosu jsou kolem -50, to je normální stav vysílání koncového zařízení.

Na obrázku 4.12 je vidět stav útoku, kdy začíná vysílat jammer, to lze rozeznat podle vyšších hodnot RSSI – -35 a velmi častého intervalu vysílání dat, a to v rozmezí 5 sekund. Čím blíž je RSSI hodnota k -30 (považuje se za silný signál) a čím častěji vysílá jammer, tím větší šance na rušení vysílání koncového zařízení. Podle toho by šlo také stanovit provedení jammingu.



Obr. 4.11: Normální stav vysílání koncového zařízení.

Obr. 4.12: Stav útoku během vysílání koncového zařízení.

Detekce jammmingu pomocí webové aplikace

Chirpstack server má k dispozici vhodné nástroje na kontrolu příchozích dat včetně aplikačního serveru se svým API, které dovoluje používat data ze serveru na jiných zdrojích. Proto bylo potřeba udělat nastavení pro externí komunikace, viz výpis 4.4.

Výpis 4.4: Nastavení externích komunikace v Dockeru.

```
1  # API a web-interface pro koncovyho uzivatele.
2  [application_server.external_api]
3  # ip:port pro svazani http serveru
4  # s web-interfacem a REST API.
```

```
5   bind="0.0.0.0:8080"
6
7   # JWT tajemstvi pro API autentizace/autorizace.
8   jwt_secret="dJIBILsFfqBOkpGIDF0Z53HT4wYmO89kUYl06B55kys"
9
10  # Povoleni hlavicky CORS.
11  # (Stejne musi byt zapnuta v browseru).
12  cors_allow_origin="{{
13  .ApplicationServer.ExternalAPI.CORSAllowOrigin }}"
14  [application_server.external_api.cors]
```

Důležitou částí je povolení CORS (Cross-Origin Resource Sharing), bez toho nelze používat API routy na externích zdrojích.

Na serveru existuje možnost kontrolovat příchozí provoz brány, což je ideální pro detekci útoku. Jsou k dispozici informace o kanálech, na kterých procházelo vysílání, jejich frekvence, šířka pásma, spreading faktor, typ dat, přesný čas přenosu, lokalita, modulace a další detaily.

V rámci bakalářské práci byla vytvořena webová aplikace pro detekci jammingu. Aplikace byla napsaná na čistém Javascriptu pomocí HTML (Hyper Text Markup Language), CSS (Cascading Style Sheets) a s použitím CSS knihovny UIkit [40]. Aplikace používá API serveru Chirpstack a musí fungovat ve spojitosti s ním.

Aplikace se skládá ze dvou částí a funguje dalším způsobem: na první stránce (viz obr. 4.13) uživatel vyplní údaje, která se týkají adresy koncového zařízení, jeho EUI (Unique Identifier), ID brány, běžný JWT (JSON Web Token) token na Chirpstack serveru (obnovuje se každých 48 hodin) a interval vysílání koncového zařízení.

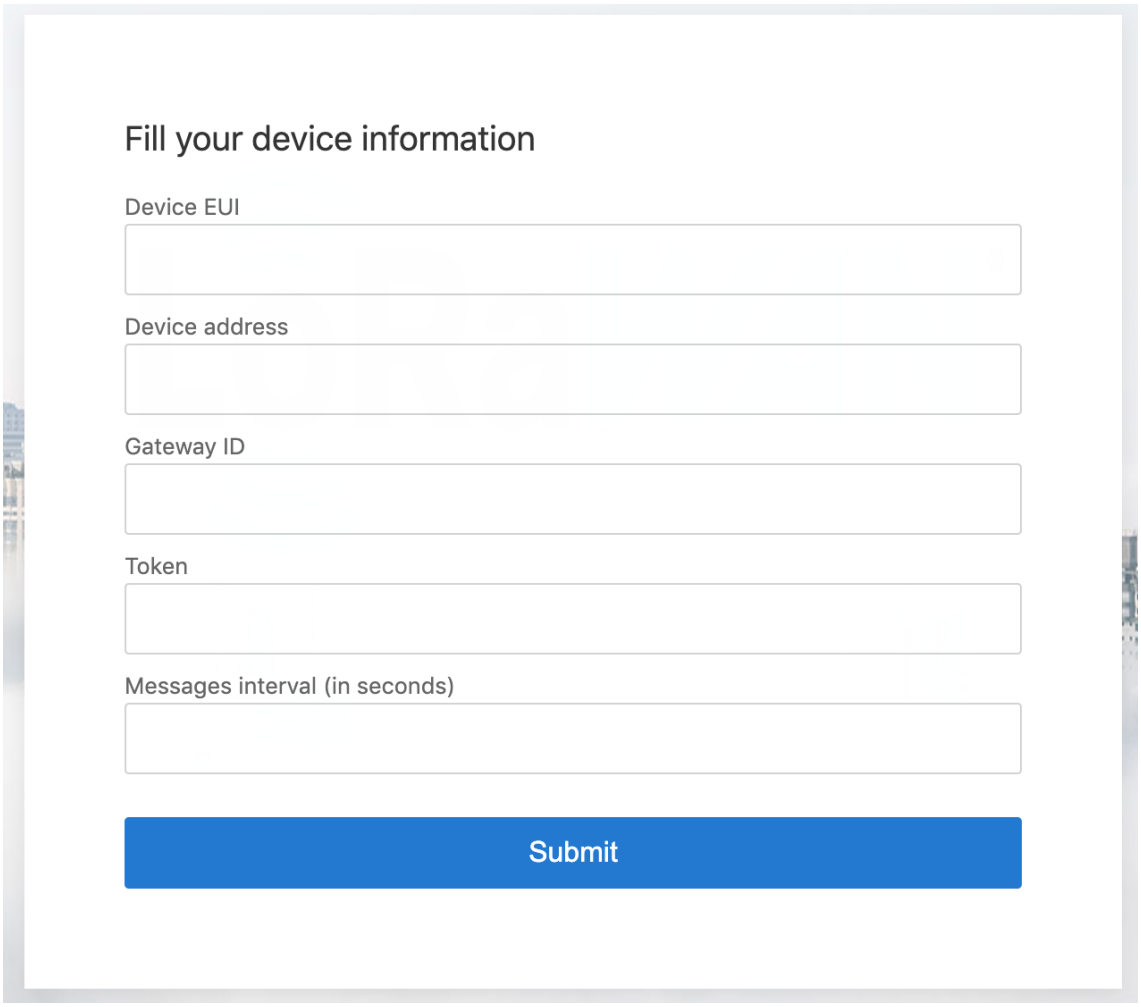
Po úspěšném vyplnění údajů bude uživatel přesměrován na další stránku, kde bude probíhat kontrola příchozích dat na bránu, viz obr. 4.14.

Během detekce se rozlišují 2 stavy: normální stav a stav útoku. V normálním stavu zařízení vysílá tak jak má, nic neovlivňuje přenos dat. Data přicházejí na bránu, kterou uvedl uživatel, aplikace označuje takové zprávy bílou barvou.

Stav útoku lze poznat podle příliš častého vysílání nebo podle příliš vysoké hodnoty RSSI. Detekce je realizována dalším způsobem: na první stránce uživatel vyplní očekávané intervaly vysílání svého koncového zařízení. Pak aplikace bude porovnávat očekávaný čas přenosu s aktuálním časem, kdy zpráva přišla. V případě, kdy zpráva přijde dříve, než je očekávaný čas přenosu, aplikace označí zprávu červenou, což znamená, že se jedná o útok. Kromě toho aplikace po každé zprávě vypočítá střední hodnoty RSSI tak, že vezme z každé zprávy tuto hodnotu, spočítá jejich celkovou sumu a rozdělí celkovým počtem přijatých zpráv, viz výpis 4.5. Detekce funguje tak, že podle střední hodnoty bude kontrolovat další zprávy. Pokud přijde

hodnota RSSI vyšší než střední, tak je velká pravděpodobnost, že na bránu přichází neověřená data a jedná se o útok. Zprávy je možné rozkliknout a podrobněji prozkoumat data, viz obr. 4.15. K dispozici jsou informace o frekvenci, šířce pásma, spreading faktoru, typu přenosu, MIC a také se zobrazuje lokalita zařízení na mapě, což umožňuje pak jednoduše stanovit, odkud vysílá původní zařízení a odkud vysílá jammer.

Aplikace je rozmístěna na „GitHub“ pomocí nástroje „GitHub pages“.

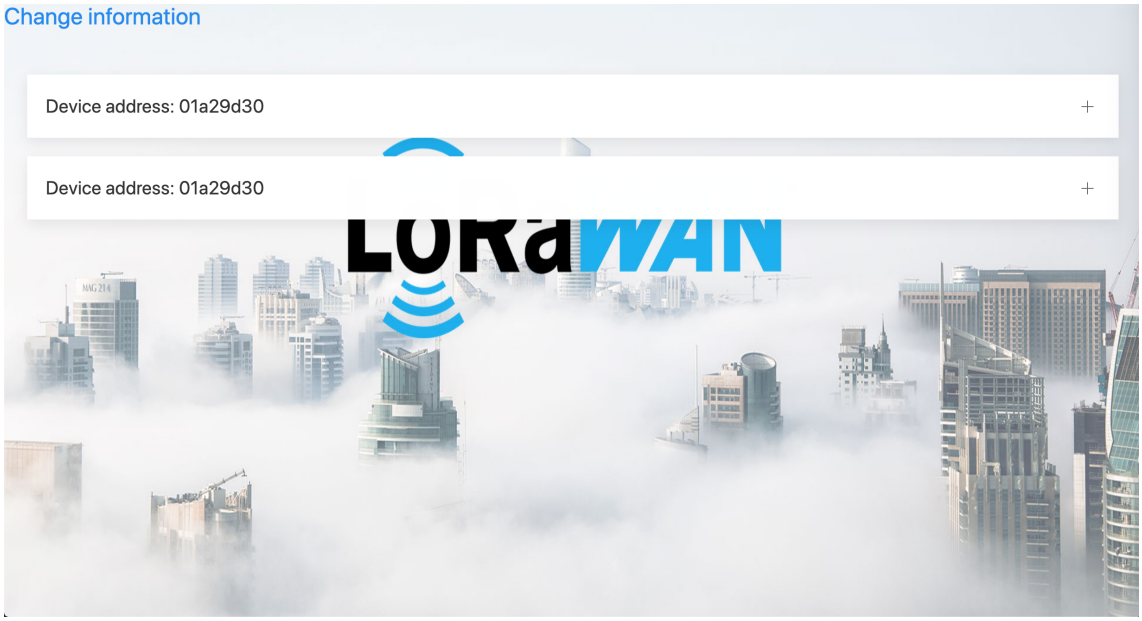


Obr. 4.13: Stránka pro vyplnění údajů.

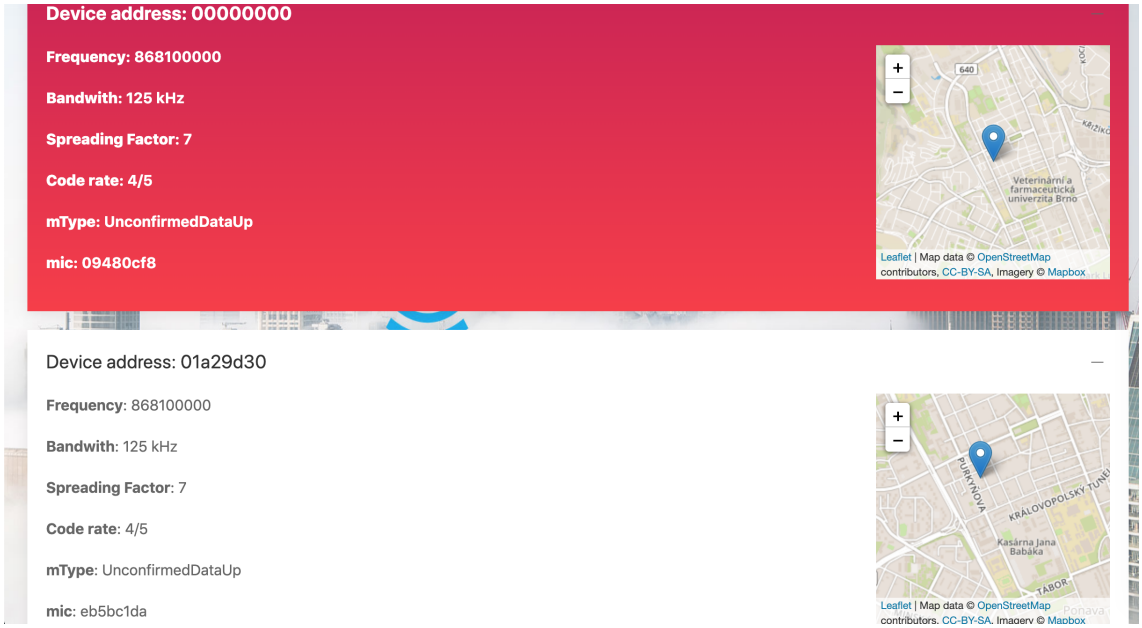
Výpis 4.5: Kód kontroly RSSI hodnoty a správného intervalu u příchozích dat.

```
1     // prevod intervalu na sekundy
2     const interval = Number(storageData.interval * 1000);
3
4     // cas prijeti zpravy
5     const newTime = new Date(time).getTime();
6     // ocekavany cas dalsi zpravy
7     const nextInterval =
8         new Date(lastDate).getTime() + interval;
9
10    // chybny cas
11    // ocekavany interval je vetsi nez cas nove zpravy
12    const notValidInterval = nextInterval > newTime;
13
14    // meni se posledni cas zpravy na ten co prisel ted
15    lastDate = new Date(time);
16
17    // pocitani stredni hodnoty RSSI
18    currentCount += 1;
19    RSSI_SUM += rssi;
20    averageRSSI = Number((RSSI_SUM / currentCount).toFixed(1));
21
22    // overeni spravnosti hodnoty RSSI
23    const notValidRSSI = rssi > averageRSSI;
24
25    // rozliseni adresy zarizeni
26    // behem aktivace a prenosu dat
27    // hodnoty jsou v ruznych mistech
28    const devAddr = macPayload.fhdr
29      ? macPayload.fhdr.devAddr
30      : macPayload.devEUI;
31
32    // kontrola sveho zarizeni
33    const validAddr = (storageData.devAddr === devAddr)
34      || (storageData.eui === devAddr);
35
36    // pokud zarizeni neodpovida tomu, co uvedl uzivatel a
```

```
37      // RSSI nebo interval jsou chybne tak se jedna o jamming
38      const notValidCheck = !validAddr
39        && (notValidRSSI || notValidInterval);
```



Obr. 4.14: Stránka detekce, normální stav.

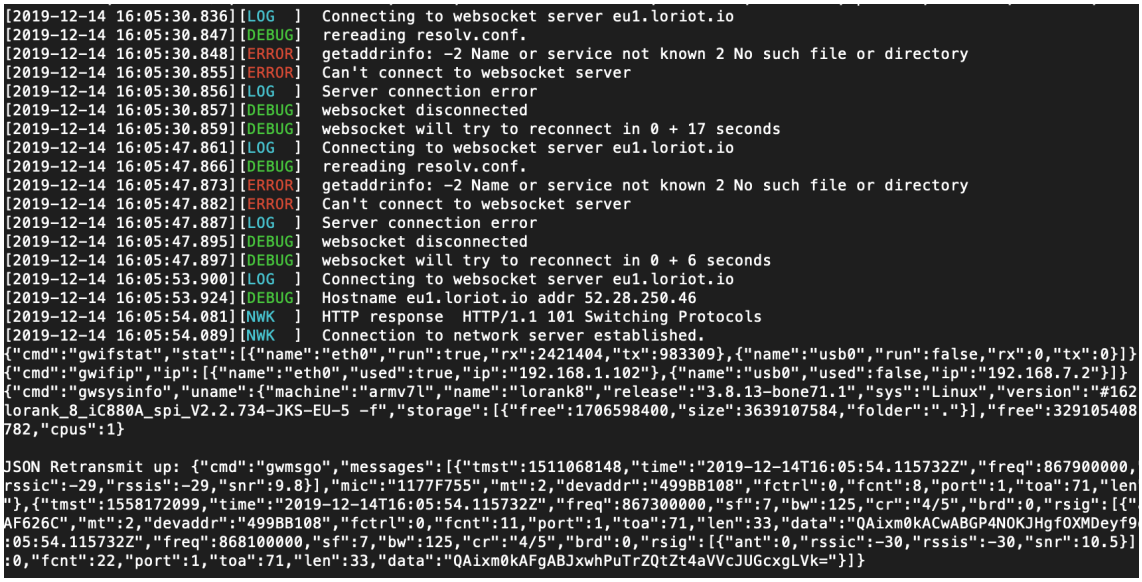


Obr. 4.15: Stránka detekce, stav útoku.

4.3.2 Detekce výpadku brány

Detekce výpadku brány většinou probíhala na Loriot serveru a také pomocí vlastní aplikace. Po zapojení sítě byly zkoumané logy na bráně, brána se připojuje na cloud pomocí SSH, následně zpracovává nutné konfigurace a začíná pravidelně přijímat LoRa rámce a posílat uplink zprávy.

Poté byl na bráně simulován výpadek sítě. Lze to lehce zjistit z logů, viz obr. 4.16. Brána se nemůže připojit k serveru a opakovaně požaduje o připojení. Po opětovném zapojení do sítě se brána automaticky připojí ke cloudu (po 3 minutách) a začne zpracovávat data dál.



Obr. 4.16: Výpadek sítě na bráně.

Detekce výpadku brány pomoci Wireshark

Pro odhalování bezpečnostních incidentů a anomálie přímo na bráně byl spuštěn program Wireshark a prozkoumán příchozí síťový provoz.

Jak již bylo zmíněno v kapitole o scénářích detekce, pro detekci výpadku brány byla použita terminálová verze Wiresharku – T-shark. Je nutné ukázat správné rozhraní, na kterém bude probíhat zachytávaní a jméno výstupního souboru. Příkaz na spuštění programu T-shark:

```
$ sudo tshark -i eth0 -w vystupni_soubor.pcap
```

Zachycená data se uloží do stanoveného souboru. Občas data přichází zkreslená, a to kvůli různým vnějším poruchám. Na odladění byl použit nástroj Pickupfix, který

zkontroluje globální hlavičku .pcap a opraví ji, pokud existují poškozené bajty. Dále začne procházet další data a kontrolovat každý bajt (jeho pořadí a velikost bloků). Pokud najde nějaký poškozený bajt nebo bajt ve špatném pořadí, tak se ho hned pokusí opravit (pokud je to možné), jinak ho přeskočí a půjde dál.

V rámci této práce byl použit SCP (Secure Copy Protokol) přenos .pcap souboru z brány na počítač, z názvu vyplývá, že jde o bezpeční přenos dat mezi dvěma uzly. Byl zachycen provoz během:

1. komunikace brány se serverem;
2. vysílání zpráv koncovým zařízením;
3. simulace výpadků sítí.

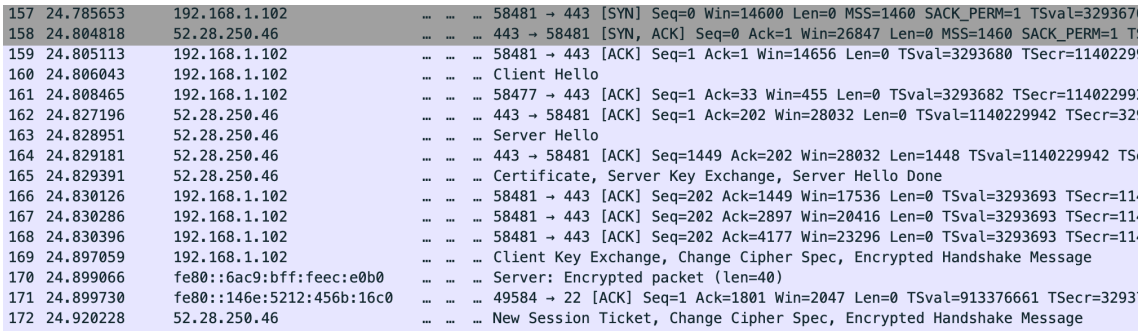
Na obr. 4.17 je vidět navázání spojení brány se serverem, na začátku se používá TCP (Transmission Control Protocol) handshake, viz obr. 4.18.:

1. SYN (Synchronize Sequence Number)) – brána (192.168.1.102) odesílá segment SYN, který informuje server (52.28.250.46) o tom, že chce zahájit komunikaci a jaké seq číslo používá.
2. SYN + ACK (Acknowledgment) – server odpovídá na požadavek klienta. Potvrzení ACK je odpověď příchozího segmentu a SYN znamená, s jakým pořadovým číslem začínají komunikovat.
3. ACK - brána potvrdí odpověď serveru a naváže spojení.

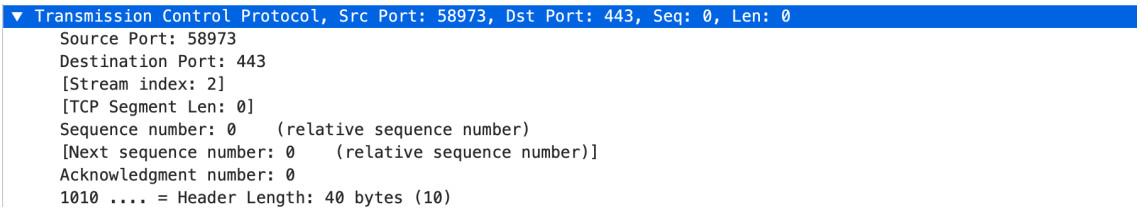
Po TCP handshaku bylo navázáno TLS (Transport Layer Security) spojení (verzi 1.2), viz obr. 4.19.

1. Client Hello: brána odešle zprávu hello obsahující verzi protokolu se seznamem šifrovacích sad a náhodným číslem klienta.
2. Server Hello: server odpovídá svým SSL (Secure Sockets Layer certifikátem, vybranou sadou šifer, náhodným číslem serveru a DH (Diffie–Hellman) parametrem. Kromě toho zpráva také obsahuje digitální podpis serveru, což potvrzuje, že server má soukromý klíč, který odpovídá veřejnému klíči s certifikátem SSL (Server Hello Done).
3. Klient dešifruje digitální podpis serveru pomocí veřejného klíče a ověřuje, zda server řídí soukromý klíč a odesílá svůj DH na server (Client Key Exchange).
4. Klient a server používají DH parametry (vyměněny mezi sebou) na výpočet premaster klíče.
5. Z náhodného čísla klienta, náhodného čísla serveru a premaster klíče pak vypočítají klíč relace a začnou komunikovat (New Session Ticket, Encrypted Handshake Message).

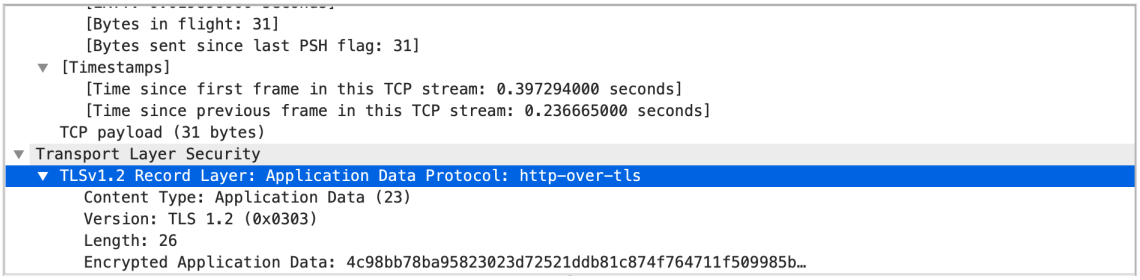
Proces navázání spojení brány se serverem během vysílání koncového zařízení je úplně stejný, rozdíl je v tom, že brána začíná vysílat data na server, viz obr. 4.20.



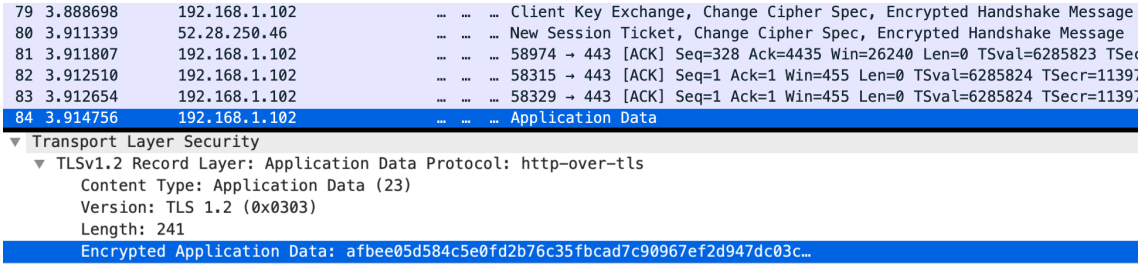
Obr. 4.17: Zahájení komunikace brány se serverem.



Obr. 4.18: Použití TCP.



Obr. 4.19: Použití verze TLS.

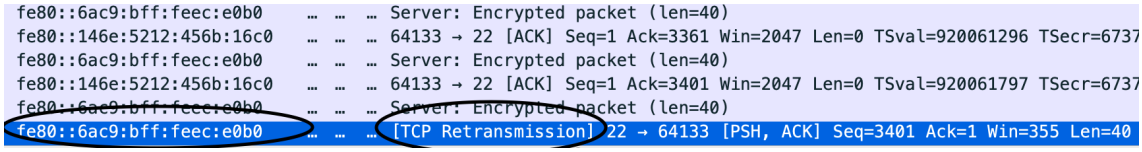


Obr. 4.20: Odesílání dat.

Data jsou posílaná v zašifrovaném tvaru (šifrovaná pomocí AppSKey). Už v tomto bodě je možné zjistit, jestli koncové zařízení přestalo vysílat data. Pokud se zařízení porouchalo nebo se odpojilo, Wireshark nezachytí žádné zprávy.

V případě plného výpadku sítě se zachytávání v Tsharku ukončí. V .pcap souboru poslední zprávou bude snaha o opětovné vysílání dat, viz obr. 4.21 (TCP

retransmission, fe80::6ac9:bff:feec:e0b0 - IPv6 adresa brány), protože TCP je spolehlivý protokol, v případě, kdy data nedojdou do cíle, vyšle je ještě jednou. Z toho se dá jednoduše pochopit, že koncové zařízení přestalo vysílat a je možné, že se porouchalo nebo na něj někdo útočí.



Obr. 4.21: Opětovné vysílání paketů.

Detekce výpadku brány pomocí webové aplikace

Během komunikace se serverem se kontroluje připojení a stav brány. Po každé zprávě byl použit timeout, který kontroloval stav brány po intervalu, který uvedl uživatel. Na začátku každého přenosu se starý interval mazal a používal se nový, to znamená, že se mohl spustit pouze když další zprávy nepřicházely a nedocházelo k jeho smazání. Jestliže zařízení přestávalo vysílat nebo se odpojovala brána, posílal se dotaz na server ohledně aktuálního stavu brány. Poté se počítal čas, kdy byla naposledy online. Jestli čas posledního připojení byl větší než poslední interval vysílání zprávy, znamenalo to, že se jedná přímo o odpojení brány (v případě odpojení koncového zařízení se žádná akce neprováděla, protože brána byla stále online), viz výpis 4.6. Následně se zobrazovalo modální okno s hláškou, že je potřeba zkontrolovat propojení nebo restartovat stránku.

Výpis 4.6: Kód kontroly výpadku brány.

```
1   // asynchronni funkce
2 async function gatewayCheck(storageData, interval) {
3   // dotaz na server a cekani na odpoved
4   const response =
5     await fetch('http://server/api/gateways?limit=1', {
6         headers: new Headers({
7           'Authorization': storageData.token,
8           'Access-Control-Allow-Origin': '*',
9         }),
10      });
11
12  // json format se parsuje v objekt
```

```
13    // dostava se policko brany
14    const { result: [ gateway ] } = await response.json();
15
16    // kdy brana byla online
17    const lastSeen = new Date(gateway.lastSeenAt).getTime();
18
19    // pocita se kolik brana je offline
20    const gatewayIsOffline = Date.now() - lastSeen;
21
22    // jestli cas v offlinu je vetsi nez interval
23    // ukaze se modalni okno a zavre se spojeni ze serverem
24    if (gatewayIsOffline > interval) {
25    // HTML kod modalu
26      const modal = `
27        <div id="modal-center" class="uk-flex-top" uk-modal>
28          <div class="uk-modal-dialog uk-modal-body
29              uk-margin-auto-vertical"
30          >
31            <button
32              class="uk-modal-close-default"
33              type="button"
34              uk-close>
35            </button>
36            <p style="text-align: center; font-size: 22px">
37              Your gateway is not responding more than
38              2 minutes. Check your connnection
39              and try to reload the application.
40            </p>
41          </div>
42        </div>`.trim();
43
44      const body = document.querySelector('body');
45      body.insertAdjacentHTML('afterbegin', modal);
46      UIkit.modal(modal).show();
47
48      socket.close();
49      console.log('Connection closed!');
50    };
51  };
```

Pro návrh sítě byly použity: koncové zařízení, brána, cloud se zabudovaným API a Chirpstack server. Na zjištění IP adresy byl použit program Wireshark a celá konfigurace probíhala pomocí SSH spojení. Stahování dat ze serveru probíhalo pomocí protokolu WebSocket a JavaScriptu.

Testování se dělo přímo na bráně kontrolováním logů systému a následným zkoumáním síťového provozu na bráně pomocí Tsharku, což je terminálová verze Wiresharku, a následného přenášení .pcap souborů s výsledky do počítače, kde byly dále zkoumané pomocí Wiresharku s GUI rozhraním. Výpadky sítě nebo jiné fyzické intervence je možné kontrolovat pomocí pravidelného monitorování a zkoumání logů nebo zachycování síťového provozu.

Zachycení provozu umožňuje odhalovat různé anomálie a útoky na síť.

Závěr

V první kapitole byly prozkoumány nejpoužívanější LPWAN technologie: Sigfox, NB-IoT a LoRaWAN se zaměřením na jejich bezpečnostní architekturu a použité šifrování.

Ve druhé kapitole byl podrobněji popsán protokol LoRaWAN vzhledem k tomu, že byl zvolen pro tuto práci a to především z důvodů možnosti vytvoření vlastní privátní sítě. Byla podrobněji zkoumána jeho architektura, klíče, šifrování, a hlavně bezpečnostní vylepšení v nové verzi protokolu oproti staré.

Ve třetí kapitole byly popsány různé hrozby a možné anomálie v síti. Následně byly předloženy návrhy a scénáře detekce, testování a odhalování nebezpečných prvků v síti včetně hrozeb na koncovém zařízení a během přenosu.

Poslední kapitola se zabývala praktickou částí. V ní byly popsány entity potřebné pro vytvoření sítě včetně brány – Lorank-8, koncového zařízení – Arduino Leonardo s IoT modulem, cloud společnosti Loriot a Chirpstack server (pro lepší pochopení problematiky byly nakonfigurování dva servery). Následně byla zprovozněna a otestovaná vlastní LoRaWAN síť. Poté byla provedena simulace bezpečnostních incidentů: jamming (rušení přenosů koncového zařízení jiným zařízením se silnějším signálem) a výpadek brány. Nakonec byla napsána vlastní webová aplikace, která byla schopna tyto incidenty detekovat. Aplikace je aktivní a funguje na doméně platformy „GitHub“ pomocí nástroje „GitHub pages“.

Literatura

[1] R. RATASUK, N. MANGALVEDHE a A. GHOSH. *Overview of LTE enhancements for cellular IoT* [online]. Hong Kong, China, 2015 [cit. 2019-10-10]. Dostupné z: https://ieeexplore.ieee.org/document/7343680

[2] ISMAIL, D., M. RAHMAN a A. SAIFULLAH. *Low-Power Wide-Area Networks: Opportunities Challenges and Directions* [online]. Conference: the Workshop Program of the 19th International Conference, 2018 [cit. 2019-10-26]. Dostupné z: https://www.gsma.com/membership/wp-content/uploads/2016/03/wp-iot-security.pdf

[3] SINHA, R SHARAN, Y WEI a Seung-Hoon HWANG. *A survey on LPWA technology: LoRa and NB-IoT.* [online]. ICT express, 2017 [cit. 2020-05-30]. Dostupné z: https://www.sciencedirect.com/science/article/pii/S2405959517300061

[4] SIGFOX. *SIGFOX EXPANDS ITS GLOBAL NETWORK TO 45 COUNTRIES* [online]. 2018 [cit. 2019-11-17]. Dostupné z: https://www.sigfox.com/en/node/656

[5] LÁSKA., Jan. *Síť pro internet věcí Sigfox pokryla celou republiku.* [online]. 2019 [cit. 2019-11-17]. Dostupné z: https://www.mobilmania.cz/clanky/sit-pro-internet-veci-sigfox-pokryla-celou-republiku-stacilo-na/-to-320-vysilacu/sc-3-a-1346186/default.aspx/

[6] SCHAFFEROVÁ, Magdalena. *Sigfox – nejpomalejší síť, kterou by měl chtít každý* Zoo Co [online]. 2017 [cit. 2019-11-18]. Dostupné z: https://www.zooco.io/blog/sigfox-nejpomalejsi-sit-kterou-by-mel-chtit-kazdy/

[7] ANTEUR, M., V. DESLANDES, N. THOMAS a A.L. BEYLOT. *Ultra narrow band technique for low power wide area communications* [online]. 2015 [cit. 2019-11-18]. Dostupné z: https://www.researchgate.net/publication/300417122_Ultra_Narrow_Band_Technique_for_Low_Power_Wide_Area_Communications

[8] LASSEN, T. *Long-range RF communication: Why narrowband is the de facto standard* [online]. 2014 [cit. 2019-11-19]. Dostupné z: http://www.mouser.com/pdfdocs/TI-Long-range-RF-communication.pdf

[9] B. VEJLGAARD, M. LAURIDSEN, H NGUYEN, I. Z. KOVÁCS, P. MOGENSEN a M. SØRENSEN. *Coverage and Capacity Analysis of Sigfox LoRa*

GPRS and NB-IoT [online]. Vehicular Technology Conference, 2017 [cit. 2019-10-12]. Dostupné z: https://vbn.aau.dk/files/253177500/LPWA_capacity_nordjylland.pdf

[10] N. MANGALVEDHE, R. RATASUK a A. GHOSH. *NB-IoT deployment study for low power wide area cellular IoT* [online]. 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2016 [cit. 2019-10-17]. Dostupné z: https://ieeexplore.ieee.org/abstract/document/7794567

[11] Qualcomm Technologies. *Performance Analysis and Uplink Scheduling for QoS-Aware NB-IoT Networks in Mobile Computing* [online]. 2016 [cit. 2019-10-17]. Dostupné z: https://www.qualcomm.cn/media/documents/files/paving-the-path-to-narrowband-5g-with-lte-iot.pdf

[12] X. CHEN, Z. LI, Y. CHEN a X. WANG. *Performance Analysis and Uplink Scheduling for QoS-Aware NB-IoT Networks in Mobile Computing* [online]. School of Computer Science, Beijing Information Science and Technology University, Beijing 100101, China, 2019 [cit. 2019-10-17]. Dostupné z: https://ieeexplore.ieee.org/document/8684953

[13] SIGFOX. *Sigfox Technical Overview* [online]. France, 2017 [cit. 2019-10-15]. Dostupné z: https://www.ismac-nc.net/wp/wp-content/uploads/2017/08/sigfoxtechnicaloverviewjuly2017-170802084218.pdf

[14] HMAC. In: *Wikipedia the free encyclopedia* [online]. 2019 [cit. 2019-10-15]. Dostupné z: https://en.wikipedia.org/wiki/HMAC

[15] O. ABDELKADER, A. OUDA a A. HAMOU. *A Survey Of Wireless Communications for IoTEcho-Systems* [online]. Canadian Conference of Electrical and Computer Engineering, 2019 [cit. 2019-10-16]. Dostupné z: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8861764

[16] BIKOS, A. N. a N. SKLAVOS. *Architecture design of an area efficient high speed crypto processor for 4g lte* [online]. 2016 [cit. 2019-10-17]. Dostupné z: https://ieeexplore.ieee.org/abstract/document/7637006

[17] DAEMEN, J. *The Design of Rijndael: AES-the Advanced Encryption Standard* [online]. Springer Science & Business Media, 2013 [cit. 2019-10-17]. Dostupné z: https://www.springer.com/gp/book/9783540425809

[18] WU, H., T. HUANG, P. H. NGUYEN, H. WANG a S. LING. *Differential attacks against stream cipher zuc* [online]. Proceedings of the 2012 International Conference on the Theory and Application of Cryptology and Information

Security, 2012 [cit. 2019-10-17]. Dostupné z: https://link.springer.com/chapter/10.1007/978-3-642-34961-4_17

[19] SINHA, A. *LoRaWAN vs NB-IoT: A Comparison Between IoT Trend-Setters* [online], 2020 [cit. 2020-05-30]. Dostupné z: https://ubidots.com/blog/lorawan-vs-nb-iot/

[20] MEKKI, K, E. BAJIC, F. CHAXEL a F. MEYER. *A comparative study of LPWAN technologies for large-scale IoT deployment* [online]. 2017 [cit. 2020-05-30]. Dostupné z: https://www.sciencedirect.com/science/article/pii/S2405959517302953

[21] *Advantages of NB-IoT / disadvantages of NB-IoT* [online]. [cit. 2020-05-30]. Dostupné z: https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-NB-IoT.html

[22] RAMACHANDRAN, G.S., F. YANG, P LAWRENCE, S. MICHIELS, W. JOOSEN a D. HUGHES. *Micropnp-wan: Experiences with lora and its deployment in dr congo* [online]. 2018 [cit. 2019-10-27]. Dostupné z: https://limo.libis.be/primo-explore/fulldisplay?docid=LIRIAS1656671&context=L&vid=Lirias&search_scope=Lirias&tab=default_tab&lang=en_US&fromSitemap=1

[23] ALLIANCE, L. *A technical overview of lora and lorawan* [online]. 2015 [cit. 2019-10-27]. Dostupné z: https://www.tuv.com/media/corporate/products_l/electronic_components_and_lasers/TUeV_Rheinland_Overview_LoRa_and_LoRaWANtmp.pdf

[24] COOPER, N. *Estimating The Service Radius* [online]. 2016 [cit. 2019-10-27]. Dostupné z: https://www.thethingsnetwork.org/community/oxford/post/estimating-the-service-radius

[25] YU, C., L. YU, Y. WU, Y. HE a Q. LU. *Uplink Scheduling and Link Adaptation for Narrowband Internet of Things Systems* [online]. IEEE Access (Volume: 5), 2017 [cit. 2019-10-27]. Dostupné z: https://ieeexplore.ieee.org/document/7842562

[26] NETWORK, T. T. *Lorawan* [online]. [cit. 2019-10-27]. Dostupné z: https://www.thethingsnetwork.org/wiki/LoRaWAN/Home

[27] LABIB, M., S. HA, W. SAAD a J. REED. *A Colonel Blotto Game for Anti-Jamming in the Internet of Things* [online]. 2015 IEEE Global Communications Conference (GLOBECOM), 2015 [cit. 2019-10-27]. Dostupné z: https://ieeexplore.ieee.org/document/7417437

[28] REYNDERS, B., W. MEERT a S. POLLIN. *Range and coexistence analysis of long range unlicensed communication* [online]. 2016 23rd International Conference on Telecommunications (ICT), 2016 [cit. 2019-10-27]. Dostupné z: https://ieeexplore.ieee.org/document/7500415

[29] VANHOEF, M. a F. PIESSENS. *Advanced Wi-Fi Attacks Using Commod-ity Hardware* [online]. In ACSAC '14 Proceedings of the 30th Annual Computer SecurityApplications Conference, 2014 [cit. 2020-03-01]. Dostupné z: https://people.cs.kuleuven.be/~mathy.vanhoef/papers/acsac2014.pdf

[30] LIPMAA, H, P ROGAWAY a D WAGNER. *Ctr-mode encryption.* First NIST Workshop on Modes of Operation, 2000. First NIST Workshop on Modes of Operation, 2000 [cit. 2020-05-30].

[31] HERSENT, O. *Lorawan specification v1.0.* Technical report LoRa Alliance Tech, 2015 [cit. 2020-05-30].

[32] MILLER, R. Lora security building a secure lora solution [online]. 2016 [cit. 2020-05-30].

[33] *Eclipse Mosquitto* [online]. 2016 [cit. 2020-05-30]. Dostupné z: https://mosquitto.org/

[34] The Things Uno [online]. 2017 [cit. 2019-12-16]. Dostupné z: https://www.thethingsnetwork.org/docs/devices/uno/

[35] SEMTECH. *Semtech SX1272MB2DAS.* Semtech.com [online]. [cit. 2020-05-27]. Dostupné z: https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1272mb2das

[36] *LoRa/LORANK 8*[online]. 2017 [cit. 2019-12-03]. Dostupné z: https://webshop.ideetron.nl/LARANK-8

[37] *ChirpStack, open-source LoRaWAN® Network Server stack.* Chirpstack.io [online]. 2016 [cit. 2020-05-27]. Dostupné z: https://www.chirpstack.io/

[38] ARDUINO. *Arduino IDE.* Arduino.cc [online]. 2003 [cit. 2020-05-28]. Dostupné z: https://www.arduino.cc/

[39] *The Things Network.* Thethingsnetwork.org [online]. 2015 [cit. 2020-05-28]. Dostupné z: https://www.thethingsnetwork.org/docs/devices/arduino/usage.html

[40] *UIkit.* Getuikit.com [online]. 2014 [cit. 2020-05-28]. Dostupné z: https://getuikit.com/

Seznam symbolů, veličin a zkratek

3GPP	3rd Generation Partnership Project
5G	Fifth Generation
ABP	Activation By Personalization
ACK	Acknowledgment
AES	Advanced Encryption Standard
AS	Access Stratum
bps	bits per second
CCM	Counter Cipher Mode with Block Chaining Message Authentication Code Protocol
CCTV	Closed Circuit Television
CMAC	Cipher-based Message Authentication Code
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
CTR	Counter
COTS	commercial-off-the-shelf
CORS	Cross-Origin Resource Sharing
CSS	Chirp Spread Spektrum
CSS	Cascading Style Sheets
DDoS	Distributed Denial of Service
DoS	Denial of Service
DH	Diffie–Hellman
DNS	Domain Name System
DPI	Deep Packet Inspection
DSSS	Direct-Sequence Spread Spectrum
ECB	Electronic Codebook
eNodeB	Evolved Node B
EUI	Unique Identifier
FEC	Forward Error Correction
FHSS	Frequency-Hopping Spread Spectrum
FNwkSIntKey	Forwarding Network session integrity key
GBA	Generic Bootstrapping Architecture
GUI	Graphical User Interface
gRPC	general-purpose Remote Procedure Calls
HMAC	Hash-Based Message Authentication Code
HTML	Hyper Text Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IDE	Integrated Development Environment

IPS Intrusion Prevention System
ISM industrial, scientific and medical
IoT Internet věcí
JWT JSON Web Token
kHz Kilohertz
LPWA Low Power Wide Area
LPWAN Low Power Wide Area Network
LoRa Long Range
LoRaWAN Long Range Wide Area Network
LTE Long-Term Evolution
M2M Machine-to-Machine
MAC Media Access Control
MAC Message Authentication Code
MCU Micro Controller Unit
MIC Message Integrity Code
MITM Man in the middle
MQTT Message Queue Telemetry Transport
NAS Non-Access Stratum
NB-IoT Narrow Band Internet of Things
NPSS Narrowband Primární synchronizace
NSSS Úzkopásmový sekundární synchronizační signál
NwkKey Network Key
NwkSEncKey Network session encryption key
OSI Open Systems Interconnection
OTAA Over-The-Air Activation
P2P peer-to-peer
PRB Physical Resource Block
QoS Quality of Service
REST Representational State Transfer
RPMA Random Phase Multiple Access
RRC Radio Resource Control
RSSI Received Signal Strength Indicator
SCP Secure Copy Protokol
SF Spreading Factor
SFTP SSH File Transfer Protocol
SNR Signal-to-noise ratio
SNwkSIntKey Serving Network session integrity key
SS Spread Spectrum
SSH Secure Shell

SSL	Secure Sockets Layer
SIM	Subscriber Identification Module
SYN	Synchronize Sequence Number)
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UICC	Universal Integrated Circuit Card
UNB	Ultra-Narrow Bandwidth
USB	Universal Serial Bus
VPN	Virtual Private Network
Wi-Fi	Wireless Fidelity

Seznam příloh

A Obsah přiloženého CD

- / ..... kořenový adresář přiloženého CD
  - xshest01.pdf ..... elektronická verze práce
  - link-web.txt ..... odkaz na aplikace
  - prikazyBrana.txt ..... příkazy používané na bráně
  - prikazyDocker.txt ..... příkaz na spouštění dockeru
  - pcapSoubory ..... adresář s zachyceným provozem na bráně
    - lora.pcap ..... normální stav přihlašování na server Loriot
    - networkFail.pcap ..... přihlašování na server Loriot při odpojení brány
  - Arduino ..... adresář s kódem pro činnost koncového zařízení
    - sendABP.c ..... sketch v jazyce C pro ABP aktivace a vysílání dat
    - sendOTAA.c ..... sketch v jazyce C na OTAA aktivace
    - TheThingsNetwork.cpp ..... knihovna v jazyce C++ s vypnutím většiny kanálů
  - chirpstackServer ..... adresář se serverem Chirpstack
    - configuration ..... adresář s konfigurací docker kontejnerů
      - application-server ..... adresář s konfigurací aplikačního serveru
        - chirpstack-application-server.toml .. konfigurace aplikačního serveru
      - gateway-bridge ..... adresář s konfigurací bridgu
        - chirpstack-gateway-bridge.toml ..... konfigurace bridgu
      - network-server ..... adresář s konfigurací síťového serveru
        - chirpstack-network-server.toml ..... konfigurace síťového serveru
      - postgresql ..... adresář se skripty na vytvoření SQL databází
        - init-chirpstack-as.sh ..... vytvoření databáze pro aplikační server
        - init-chirpstack-ns.sh ..... vytvoření databáze pro síťový server
    - docker-compose-env.yml ..... proměnný okolí pro docker-compose
    - docker-compose.yml ..... konfigurační soubor pro kontejnery
    - package-lock.json ..... závislosti pro server
  - detectionApp ..... adresář s webovou aplikací pro detekci jammingu
    - assets ..... adresář s obrázky
      - LoRaWAN-cover ..... obrázek na pozadí stránky
    - pages ..... adresář se stránkami
      - detection ..... adresář se stránkou detekci
        - detection.css ..... CSS styly pro stránku detekci
        - detection.html ..... HTML dokument stránky detekci
        - detection.js ..... JavaScript soubor s logikou pro stránku detekci
      - welcome ..... adresář s přihlašovací stránkou
        - welcome.css ..... CSS styly pro přihlašovací stránku
        - welcome.html ..... HTML dokument přihlašovací stránky
        - welcome.js ..... JavaScript soubor s logikou pro přihlašovací stránku
    - package.json ..... závislosti pro aplikace